A PROTOTYPE EXPERT SYSTEM FOR TECHNICAL ORDER ACQUISITION

THESIS

James F. Harvell
Captain, USAF

AFIT/GSM/LSY/88S-11

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSM/LSY/88S-11

A PROTOTYPE EXPERT SYSTEM FOR TECHNICAL ORDER ACQUISITION

THESIS

James F. Harvell
Captain, USAF

AFIT/GSM/LSY/88S-11

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information is contained therein. Furthermore, the views expressed in the document are those of the author and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the United States Air Force, or the Department of Defense.

AFIT/GSM/LSY/88S-11

A PROTOTYPE EXPERT SYSTEM FOR TECHNICAL ORDER ACQUISITION

THESIS

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Systems Management

James F. Harvell, B.S.

Captain, USAF

September 1988

Preface

The purpose of this research was to develop a prototype expert system for technical order acquisition. This prototype system serves as an initial effort in applying expert system technology to the functional area of acquisition logistics in weapon system acquisition. The effort required to build, evaluate, and maintain an expert system or systems for acquisition logistics borders on monumental. However, the eventual benefit to acquisition logisticians is a powerful management tool that is based on the years of experiences of acknowledged experts.

This research needs to be continued. The prototype system needs to be fielded, refined, and expanded. Hopefully, the exposure of the prototype will spark the application of expert system technology to other segments of acquisition logistics.

In building the prototype system and writing this thesis, I received help from several people who deserve much more than the simple gratitude I can express here. Mr. O. J. Frazier served as the TO expert for building the prototype which reflects his knowledge of TO acquisition. Mr. Riley Gust, Mr. Art Mungia, Mrs. Marie Rotert, and TSgt Michael Mires evaluated the prototype and provided me with excellent support. To these people, I wish to thank you all. Maj Thomas Triscari's mastery of the art of subtle and not so subtle persuasion is probably the primary reason this thesis was completed. I am thankful for his persuasive abilities. Above all, my deepest gratitude is reserved for my wife, Gloria, and my daughter, Jessica. Their quiet sacrifice thunders with devotion to our family.

James F. Harvell

Table of Contents

Page

List of Figures

List of Tables

Abstract

The purpose of this research was to build a prototype expert system for technical order (TO) acquisition. The research focused on the following six areas: the applicability of expert system technology to TO acquisition tasks; the required resources, participants, goals, and problem characteristics for the prototype: the key concept and relations in the selected domain of TO acquisition; the appropriate knowledge representation scheme and development tool; the required data structures and control strategies; and the competency and utility of the prototype. The research drew conclusions on each of these six areas.

The research found that both planning and executing a TO acquisition program are suitable tasks for expert system technology. The required resources for developing a prototype system for TO acquisition are pertinent literature on TO acquisition, a willing TO expert, approximately ten weeks of the knowledge engineer's time, and access to computing facilities and expert system development tools. The participants required to develop a prototype system are a TO expert(s) to build the prototype, additional TO experts to validate the prototype's performance, and typical end-users to assess the prototype's utility. The appropriate goals for a prototype system for TO acquisition are formalizing an informal set of procedures and distributing scarce expertise to inexperienced TO managers in developing the contractual documentation for acquiring TOs. The problem domain was characterized by the use of the weapon system program's attributes to provide TOs with the most current and accurate information in a timely manner. The key concepts in TO acquisition are the program phase, maintenance concept,

complexity of the technology, requirement for source data, number of TOs being acquired, type of program, and the classification of the TOs. These key concepts relate to the contractual documentation for TO acquisition. A rule based scheme is the most appropriate knowledge representation scheme for TO acquisition. The most appropriate tool for developing the prototype is VP-EXPERT. The data structures should represent the key concepts. The prototype's control strategy is backward chaining through the rule base. The prototype is competent in determining the contractual documentation for TO acquisition. From the standpoint of the prototype's utility, the prototype is judged as useful in helping inexperienced TO managers in preparing the contractual documentation and in training.

The recommendations address the suggested use of the prototype and further research. The prototype should be sent to ASD for use on a day-to-day basis by TO managers in the system program offices. Such day-to-day usage would serve to test the system further and discover its strengths and weaknesses. Research into expanding the prototype system for TO acquisition could include adding other TO experts' knowledge to the knowledge base, tailoring sections four and five of the TMCR, and expanding the explanations for the actions and conclusions of the prototype. On a larger scale, the application of expert system technology to other areas of weapon system acquisition is warranted. Most of the functional areas of weapon system acquisition are prime candidates for expert system applications. Prototype expert systems for those areas should be developed.

A PROTOTYPE EXPERT SYSTEM FOR TECHNICAL ORDER ACQUISITION

I. Introduction

Overview

This chapter discusses the use of decision support systems in weapon system acquisition management. Next, the research problem, the purpose of the research, and the justification for the research are stated. The chapter concludes a declaration of the scope and limitations of the research.

Background

The high profile of artificial intelligence (AI) technology has captured the interest of the Air Force. This interest has not been limited to using the technology in weapon systems. In fact, the subject of a recent thesis completed by Capt Robert J. Hammell at the Air Force Institute of Technology is an application of expert system technology (a subset of AI) to developing acquisition strategies for procurement programs. In addition, the Defense Systems Management College (DSMC) has an on-going effort to develop decision support systems for different facets of acquisition management.

Capt Hammell's thesis demonstrated the promise of expert system technology for developing acquisition strategies. The thesis examined the need for expertise in developing acquisition strategies, and the reasons for the inadequacy of conventional programming. The remainder of the research was devoted to developing and evaluating the prototype system for developing acquisition strategies. Capt Hammell's research

concluded that an expert system could "provide significant benefit" to those charge with developing acquisition strategies (14:viii).

The DSMC is an educational and research institution that is dedicated to the improvement of Department of Defense (DoD) weapon system acquisitions. Acquisition management education is the DSMC's primary mission. The DSMC is charged with educating DoD's acquisition personnel in "program management policies, philosophies, skills, and techniques necessary for the effective and efficient execution" of acquisition projects. The DSMC's educational mission is complemented by DSMC's research in applied management science. The information that is a natural product of DSMC's educational and research efforts is disseminated by the DSMC to the acquisition community. One of the DSMC's research projects is an application of decision support system technology to defense acquisition management. This research project is titled the Program Manager's Support System (PMSS) (6:1-2).

The completed PMSS will assist program managers and their staff in making decisions and in executing programs effectively and efficiently (6:iii). The PMSS will be an integrated software system for use in system program offices on microcomputers. Also, the PMSS will integrate functional areas (such as engineering, logistics, configuration control, etc.), give program alternatives, and assess impacts from program management decisions. Program management decisions will still be the sole domain of the program manager, but the PMSS will give the program manager the ability to rapidly assess the impacts of a decision across the entire spectrum of functional areas (6:iii-iv,3,7).

The PMSS will provide this capability through two major components. The first component is a collection of functional modules (each module can act as a stand-alone program) that are designed to assist in areas

such as planning, acquisition document generation, budgeting, production planning, and configuration management. The second component is the integrated PMSS that will integrate the functional modules. The integrated PMSS will look across and within functional areas to give program managers a means to assess the impacts of program decisions and alternatives. The functional modules are in various stages of development and the integrated PMSS is to be completed in fiscal year 1988 (6:iii-iv,20).

One of the functional modules in development is the Automated Program Planning and Documentation Module (APPDM). The APPDM will assist program managers in planning activities and documenting those plans in the Program Management Plan, Production Strategy/Plan, Test and Evaluation Master Plan, System Engineering Management Plan, Risk Analysis and Management Plan, and the Integrated Logistics Support Plan (ILSP). Focusing on the ILSP in the APPDM's abilities shows that the module provides a means to track the progress of the ILSP's development and a boiler plate document for tailoring to a program's needs. Beyond helping the program manager to generate the ILSP, the PMSS has little ability to assist the program manager in actively managing the logistics elements of a weapon system program (a future module has been identified for initial and replenishment spares) (6:56). The logistics elements, collectively referred to as supportability, have been the subject of intense scrutiny by the acquisition community in the last few years.

This intense scrutiny resulted in increased emphasis on acquisition logistics, which is the area of program management that deals with weapon system supportability. Weapon system supportability has gained equal status with the traditional measures of program success: technical performance, cost, and schedule. The elevation of supportability to

equal status with technical performance, cost, and schedule is a direct result of the realization that weapon systems that cannot be maintained and supported are ineffectual weapon systems. Therefore, the acquisition community placed increased emphasis on weapon system supportability and the integrated logistics support (ILS) program that ensures a new weapon system's supportability.

The ILS program gives management an approach to develop support requirements, to integrate support requirements into weapon system design, and to acquire the support at an affordable life cycle cost. The objective of the ILS program is to deliver a logistically supported and supportable system (10:1). The program manager is responsible managing the ILS program and for achieving the ILS program objective (7:6). However, the program manager usually delegates ILS program management functions to a deputy program manager for logistics (DPML, for major programs) or an integrated logistics support manager (ILSM, for less than major programs) (10:5,7).

The DPML or ILSM's functions concern management of the ILS elements which are maintenance planning, manpower and personnel, supply support, support equipment, technical data, training and training support, computer resources support, facilities, packaging/handling/storage and transportation, and design interface (logistics related design parameters). With regards to technical data, the DPML or ILSM is concerned with planning for and managing the development of technical orders (TOs). The importance of TOs cannot be overstated. TOs describe the procedures for operating and maintaining Air Force systems and equipment (7:4). To acquire TOs, the DPML or ILSM is involved in budgeting, defining contractual requirements, scheduling, testing, and integrating the support efforts of the supporting command and

participating commands (10:5,11-13). From the description above, the high degree of complexity and difficulty in managing a TO acquisition, even for an experienced acquisition logistician, is evident. Even a very experienced DPML or ILSM must rely on a few experts in TO acquisition for guidance.

The value of these experts is obvious considering that the acquisition logistics functional organization has a large number of inexperienced junior officers and junior civilians in need of training and guidance. The guidance provided by these experts is usually in a particular ILS element which is a narrow problem domain. Such a narrow problem domain makes the application of current expert system technology to TO acquisition feasible (23:8).

Statement of Problem

The acquisition logistics functional organization at Aeronautical Systems Division (ASD) has many inexperienced junior officers and junior civilians who could benefit from ASD's acquisition logistics experts' knowledge. The problem is that no system exists at ASD which captures the TO acquisition (or any other ILS element) experts' knowledge for use by the inexperienced DPML or ILSM.

Purpose

The purpose of this research is to develop a microcomputer based prototype expert system for acquisition logistics at ASD. During the development of the prototype, the research will attempt to answer the following questions.

1. What TO acquisition tasks are suitable for expert system application?

2. What are the required resources, necessary participants, appropriate goals, and problem characteristics for a prototype system for TO acquisition?

3. What are the key concepts and relations used in TO acquisition?

4. What is the appropriate knowledge representation scheme and tool for developing the prototype system?

5. What data structures and control strategies are required in the prototype system?

6. How competent and useful is the prototype?

Justification

In July 1987, the commander of ASD, Lt Gen William Thurman, authorized the establishment of the ASD Artificial Intelligence Office to expedite the transition of artificial intelligence (AI) technology from the laboratories to the system program offices. Concurrently, Lt Gen Thurman directed a study to determine the aspects of acquisition management that could benefit from AI technology (29). The study was conducted by Lt Col Gregory Parnell and Capt Thomas Triscari, both of the Air Force Institute of Technology faculty. One of the major findings and recommendations of the study is as follows:

> ASD functional organizations have the expertise and the incentive to develop system acquisition management expert systems. The report identifies potential expert systems and recommends that a knowledge engineering problem assessment be performed for each potential expert system [23:1].

Although the study did not specifically address acquisition logistics, the study states that "expert systems could be identified in . . . acquisition logistics" (23:9).

Expert systems for TO acquisition could alleviate the burden on the experts who as matrix support are often supporting more than one program. This results in competing demands on the experts that limit the amount of guidance they can provide DPMLs and ILSMs. Another problem that an expert system could solve is the void that is left when an expert retires or is transferred by capturing that expert's knowledge for future use.

Finally, the Packard Commission's recommendation that the number of people involved in weapon system acquisition be reduced lends credence to the effort to increase effectiveness and efficiency through expert systems (26:55).

Scope and Limitations

The scope of the study is to develop a prototype expert system to assist in one task of TO acquisition. The task will be selected based on the criteria of amenability to, justification for, and appropriateness of expert system application to that element. The study will be limited to developing a prototype expert system for one task because of the time constraints on the research. Also, the study will be limited to ASD because of the close proximity and the high amount of face-to-face interaction required between the researcher and the TO acquisition expert(s).

Summary

This chapter presented background material on the current efforts in applying AI technologies to acquisition management, acquisition logistics, and TOs. Following the background material, this chapter stated the problem underlying the proposed research. The chapter concluded with the purpose of, justification for, and scope and limitations of the research. The next chapter is a review of the literature pertinent to this research.

II. Literature Review

Overview

This chapter reviews some of the current literature written about expert systems. In order to provide a common foundation for this research, the first nine sections of this chapter pertain to expert systems and the remaining two sections to TO acquisition. The chapter's first three sections address the evolution, types, and applications of expert systems. The next two sections cover the structure and characteristics of expert systems. The last four sections on expert systems cover the practical knowledge required for developing expert systems. This practical knowledge is divided into knowledge acquisition, knowledge representation, expert system development tools, and expert system development methodology. The two sections on TO acquisition relate to the TO acquisition process and the process's problems.

Evolution of Expert Systems

The evolution of expert systems traces its beginnings to the first efforts by British scientists at the end of World War II to develop what is now called a computer. The British scientists were interested in developing a machine for general purpose problem solving using logical operators to manipulate symbolic data. Eventually, this approach succumbed to the more practical American approach which focused on using arithmetic operators. However, the British approach was not abandoned entirely. A small group of computer scientists continued working on symbol manipulation using computers. Simultaneously, researchers in the fields of formal logic and cognitive psychology were attempting to

develop computer programs to mimic human problem solving. The merging of the work on computer manipulation of symbols and the work on human problem solving resulted in the creation of what is now popularly called artificial intelligence (AI) (15:2-3).

AI's formative years were marked by the increased availability of computers and advances in understanding the psychology of information processing (15:4). Initial research in AI was guided by the belief that supercomputers with a few laws of reasoning would be capable of solving a wide range of problems (16:7). Consequently, the goal of the early AI researchers was to develop general problem solving approaches that could be used to solve a broad classes of problems (30:3; 13:1). During the 1960s, this goal proved to be very difficult to attain as evidence mounted that general purpose problem solving strategies were unable to cope with complex problems. The evidence indicated that a change in the focus of AI to more narrowly defined problems offered a better chance for success (30:4; 16:7).

The redirected focus of AI to building systems for solving problems in narrowly defined problem domains resulted in the emergence of several expert systems in the mid-1970s (16:7; 12:16). This shift away from general problem solving systems to systems with a large amount of knowledge about a specific problem is regarded as a key breakthrough in AI research (15:5). However, these early expert systems were limited in their success because of the researchers' concentration on knowledge representation schemes and search techniques (30:4; 16:7). Recognizing the lessons from these early expert systems, Professor Edward Feigenbaum provided the next breakthrough in the concept of AI, in general, and expert systems, in particular. His conceptual breakthrough is that a program's problem solving power comes from the knowledge it possesses,

not from the particular formalisms and inference schemes it uses (30:4; 16:7).

Feigenbaum's breakthrough provided the foundation for expert systems developers to build upon. Also, advances in the power and speed of computer hardware were instrumental in the current proliferation of expert systems (12:15; 15:3). Expert systems development was, at first, an art rather than a science. Today, the development of expert systems is a better defined process (30:5). Therefore, after years of research, expert systems have moved out of the laboratories and into practical applications (12:16-17). Expert systems represent the most practical and accepted application of AI (13:3).

Description of an Expert System. A universally accepted description of an expert system does not exist. However, a common theme does appear among the varying definitions and descriptions of expert systems. That common theme is that an expert system is a computer program designed to emulate the performance of an expert in some problem domain (30:8; 15:5; 13:3; 16:169; 12:2).

Waterman defines an expert system as "a computer program using expert knowledge to attain high levels of performance in a narrow problem area" (30:11). Similarly, Hayes-Roth, Waterman, and Lenat describe an expert system as "a computer system that achieves high levels of performance in task areas that, for human beings, require years of special training and education" (16:400). Buchanan and Shortliffe state that expert systems exhibit three characteristics: the ability to solve complex problems like an expert, the ability to be understood, and flexibility (4:3). Harmon and King cite Feigenbaum's definition of an expert system as "an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to

require significant human expertise for their solution" (15:5). Each of these definitions and descriptions focuses on expert systems as computer programs that perform at a high level in some problem domain.

Advantages and Benefits of Expert Systems. Expert systems offer several advantages over human expertise. One advantage is the permanence of expert systems' expertise. Human expertise is often subject to degradation if the expertise is not exercised regularly. Another advantage is the ease of transferring expertise by copying the computer program. Transferring expertise from one human to another is a long, laborious process. The process of documenting human expertise is very difficult and time-consuming. Expert systems offer the advantage of documenting the human expertise through the knowledge engineering process. Also, expert systems produce more consistent results than human experts, because expert systems are not susceptible to distractions. A final advantage is the relative low cost of an expert system as compared to the cost of developing and maintaining human expertise (30:12-13).

The benefits of expert systems include improved productivity, preservation of valuable knowledge, and improvement in understanding and learning. Productivity can be improved by using expert systems that can solve more problems in a shorter amount of time than a human expert. An expert's valuable knowledge can be preserved in an expert system for use even if the expert leaves the organization. Expert systems improve understanding of how the human expert reasons by explicitly stating the expert's problem solving techniques. In addition, expert systems improve the learning of novices in a problem domain by familiarizing the novices with the domain subject matter (12:4-6).

Despite the advantages and benefits of expert systems, some researchers believe the most significant contribution of expert systems

is really a by-product of expert system development. This contribution is the codification of knowledge. The codification and accumulation of knowledge that is accessible and explicit has value for human culture beyond the computer program (30:7; 16:27-28).

Types of Expert Systems

Expert systems are categorized by the type of problem addressed by the system. Consequently, ten types of expert systems have been identified. The ten types of expert systems are interpretation, prediction, diagnosis, design, planning, monitoring, debugging, repair, instruction, and control (30:33; 16:13-15; 12:34-39; 13:44).

Interpretation systems infer descriptions based on observed data. The data is assigned symbolic meaning that describes the situation. Typical interpretation systems are surveillance, image analysis, and chemical analysis (16:13).

Prediction systems infer the outcomes of a situation. The situation is sometimes represented by a model that interfaces with the prediction system. Typical problems addressed by prediction systems are crop estimates and oil demand estimates (30:34).

A diagnostic expert system uses observed data to make inferences about malfunctions. The cause(s) of the malfunction is then suggested by the system. Diagnosing large complex systems, especially electronics, and medicine are common problem areas for diagnostic systems (12:35-36).

Design systems are given a set of constraints to use in configuring a set of objects. Current applications of design systems are molecular biology and microelectronics. Design systems are closely related to planning systems (13:40).

Planning systems specialize in problems where objects perform functions. This type of system infers the effects of the planned actions. Planning problems include project, route, and military planning (16:14-15).

The function of monitoring systems is to compare actual behavior with expected behavior. Monitoring systems contain contextual and time elements for interpretation of the observed behavior as normal or deviant. Monitoring systems have uses in problem areas such as nuclear plant and intensive care monitoring (30:36).

Debugging systems recommend corrective actions for problems. Powerful debugging systems design solutions and predict the effectiveness implementing the solution. Suggesting maintenance actions for electrical cable and aircraft repairs is an example of a debugging system's functions (12:36; 13:40).

After a problem has been diagnosed and debugged, repair systems provide plans to execute the prescribed remedy. Expert systems as repair systems are beginning to impact automotive, avionic, and computer repair fields (16:15).

Instructional systems diagnose and debug student behavior by modeling the student's knowledge and developing plans to correct any deficiencies in that knowledge. The system then executes those plans through interaction with the student. Instructional systems are used by the Navy to train personnel in the operation of steam propulsion plants (30:37-38).

The most ambitious type of expert system is the control system. A control system manages the behavior of the entire system. Therefore, the control system has to interpret the situation, make predictions, diagnose problems, plan solutions, and monitor the execution of the solutions.

Control systems are being applied to business management, battle management, and air traffic control.

Applications of Expert Systems

Expert systems have found applications in a diverse group of problem domains. Currently, over one hundred seventy expert systems are in use in sixteen application areas (13:vi-xi). The most active application areas have been chemistry, computer systems, electronics, engineering, geology, medicine, and the military (30:40).

The first expert system in chemistry was DENDRAL. DENDRAL infers a compound's molecular structure from mass spectral and nuclear magnetic response data. The DENDRAL project was started in 1965 and is still used regularly by chemists (30:40; 16:51). DENDRAL originated the manipulation of expert heuristic information in a problem solving form, a key idea in expert systems (16:38). Other expert systems in chemistry work on inferring molecular structures, synthesizing organic molecules, and planning experiments (30:40).

A typical computer system expert system is XCON. XCON designs the configuration of Digital Equipment Corporation's VAX computer systems. Beginning as a research project in the 1970s, XCON has matured to the point of being a commercial system (30:40; 13:251). Other problems in computer systems that are the subject of expert system work are mainframe monitoring, program conversion, automatic programming, and data communications (12:61-63).

SOPHIE is an instructional system that teaches electronics troubleshooting. SOPHIE is considered a landmark system because of its use of a semantic network for knowledge representation and reasoning (16:41). Expert systems in electronics have focused on troubleshooting and circuit design (30:40-41; 12:58).

Current work in engineering expert systems is in fault diagnosis and instruction of control processes. For example, General Electric is using an expert system named DELTA to isolate malfunctions in diesel locomotives. Another example is the expert system REACTOR, which monitors nuclear reactor operations (30:41-42).

The first expert system for geology was PROSPECTOR. PROSPECTOR provides expertise for finding ore deposits (30:44). PROSPECTOR's contribution to expert system evolution was its ability to accommodate new knowledge (16:41). Geological expert systems in use address the problems of well analysis and malfunctions in drilling operations.

More expert systems have been developed for medicine than any other field. One of the earliest and most famous expert systems is MYCIN (30:40; 12:54). MYCIN aids medical students in learning how to diagnose and treat infectious blood diseases (30:44; 16:53). MYCIN's key feature is it's use of certainty factors for probabilistic reasoning (16:39). Other medical expert systems interpret test data, diagnose disease, recommend treatment, and instruct medical students (30:44).

The military's interest in expert systems has been in interpretation, prediction, and planning systems. The first application of expert system technology to the military was HASP/SIAP, which identifies ships by interpreting sonar data (30:45). The armed forces are using or working on expert systems for interpretation of sensor data, prediction of armed conflicts, tactical planning, and equipment maintenance (30:45; 12:64).

Characteristics of Expert Systems

An appropriate introduction to the characteristics of an expert system is a comparison of the difference between conventional programs

and expert system programs. The fundamental difference is that conventional programs use data while expert system programs use knowledge. Knowledge allows expert systems to pursue an inferential process as opposed to the repetitive processing of conventional programming. Conventional program processing requires the effective manipulation of large data bases. However, expert system programs require the manipulation of large knowledge bases (30:24; 13:34). The knowledge base includes heuristics while conventional programs rely on algorithms (30:24; 12:9-10; 13:34). An algorithm is designed to give a correct answer every time; however, an expert system is designed to emulate a human expert. Since human experts may make mistakes, expert systems may make mistakes as well. (30:29; 13:32). Conventional programs do not exhibit the characteristics that distinguish expert systems.

Expert systems exhibit the characteristics of expertise, symbolic reasoning, depth, and self-knowledge. Expertise refers to the system's ability to attain a high level of performance in the least time possible. This implies that expert systems must be skillful in applying its knowledge to produce high quality solutions using efficient search techniques. These search techniques are representative of how the human expert arrives at a solution quickly. In order to perform like a human expert, an expert system needs robustness, depth of domain knowledge and breadth of knowledge. The breadth of knowledge gives the expert system some general problem solving strategies to use when given incomplete information (16:43-45; 30:25).

The type of information that expert systems use to solve problems is symbolic. In problems suitable for expert systems, human experts represent the problem concepts through symbols. These symbols are then manipulated to arrive at a solution. Therefore, an expert system must be

able to manipulate symbols. Thus, knowledge representation (the choice, form, and interpretation of the symbols) is a very important concept (16:45-46; 30:26). Another important concept in symbols is the expert's ability to reformulate the problem and symbols to arrive at a solution more efficiently. Most expert systems do not have this problem reformulation ability (16:47; 30:26).

Expert systems have the ability to operate effectively in a narrow problem domain that is complex and difficult. This characteristic is referred to as depth. Since depth is linked to the problem domain, the level of problem simplification is an indication of the depth of an expert system. A low level of problem simplification represents a real-world domain where solutions can be applied in a practical manner. In contrast, a domain which has a high level of problem simplification produces solutions that are of theoretical interest only. Another indication of the depth of an expert system is the size and complexity of the possible intermediate and final solutions (16:47; 30:26-27; 13:31).

Self-knowledge refers to an expert system's ability to reason about its own processes. This self-knowledge is sometimes called metaknowledge, which means knowledge about knowledge (30:27-28; 13:31). The first practical consequence of self-knowledge is the ability to explain how the system formulated its solution. Most expert systems have such an explanation facility. However, self-knowledge's benefits extend beyond explanation. Self-knowledge offers the possibilities of creating the rationale behind rules and tailoring the explanation to the audience (16:48-49; 30:27-29; 13:31).

Structure of Expert Systems

The structure of every expert system is not identical. The most fundamental description of an expert system's structure divides an expert

system into a problem solving component and a support component. The function of the problem solving component is self-explanatory, and the function of the support component is to facilitate user interaction with the system (30:8; 13:10). However, every expert system does have the following major components: knowledge base, inference engine, data base, and user interface (see Figure 1) (12:99).

The heart of any expert system is the knowledge it contains. The knowledge can be rules or facts (30:16). The rules and facts pertaining to the problem domain are contained in the knowledge base (16:19; 30:18; 12:99; 4:4). A key feature of the knowledge base is the separation of domain knowledge from the general problem solving knowledge of the inference engine (30:18; 12:99). An expert system with its knowledge organized in this manner are called knowledge-based systems. The vast majority of expert systems are knowledge-based systems (30:18).

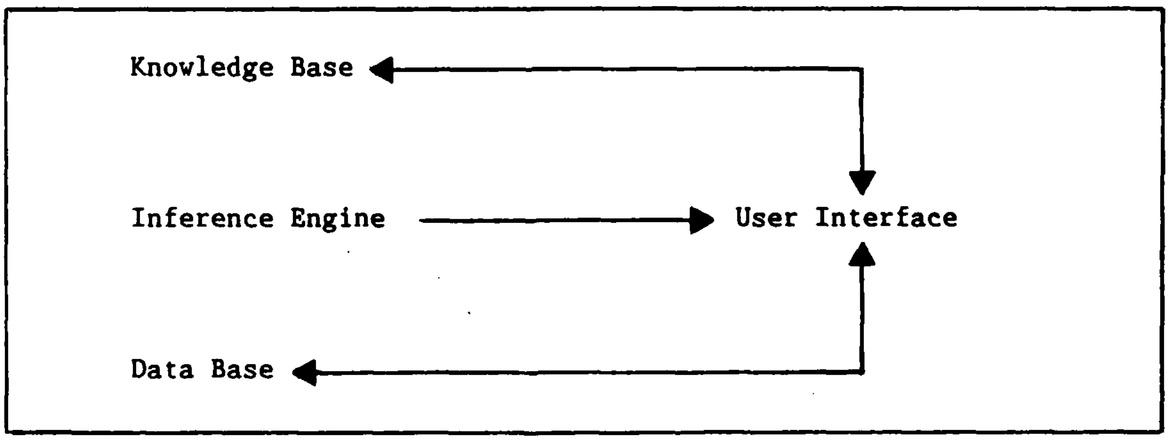
There is no simple, general way to describe an inference engine because of the many different forms it can take (30:19; 4:4). The inference engine's two main functions are inference and control (12:101). The inference engine performs these functions through the scheduler and the interpreter. The scheduler controls the order in which rules or facts from the knowledge base are applied. The interpreter determines how to use the domain knowledge to draw inferences (30:22-23; 16:18). Also, an inference engine may have a checking capability to enforce consistency and completeness in the knowledge base as it is modified (21:75).

The data base is analogous to a scratch pad. During system execution, the data base records inputs, intermediate conclusions, and outputs (12:102). A particular type of data base is called a blackboard. The blackboard records intermediate conclusions and

decisions concerning plan, agenda, and solution elements. Plan elements describe the system's approach to solving the problem. The agenda elements records the actions to be taken. Solution elements contains the candidate hypotheses, decisions that have been made, and description of how the hypotheses and decisions relate to each other (16:16-18).

The user interface works with the inference engine and knowledge base to allow the user and the expert system to communicate (12:102). The interface bridges the gap between the user and the other components of the system (16:16).

Additional subprograms that are components of some expert systems include an explanation facility and a knowledge input subsystem (12:103-104). An explanation subsystem explains the steps taken to arrive at a solution (30:30; 12:103). In addition, an explanation subsystem should justify the steps taken and answer questions about domain terminology and the intent of an action (28:196). A knowledge input subsystem is particularly useful in a domain that is dynamic, because the knowledge base can be modified easily (12:104).



(16:17)

Figure 1. Components of an Expert System

Knowledge Acquisition

Hayes-Roth, Waterman, and Lenat define knowledge acquisition as "the transfer and transformation of problem solving expertise from some knowledge source to a program" (16:129). Considering the previous assertion that the power of an expert system comes from the knowledge it possesses, acquiring knowledge is an important aspect of developing an expert system (16:128). While knowledge can come from many sources (periodicals, data bases, personal experience, etc.), the emphasis in knowledge acquisition has been placed on the human expert. The time and effort required to extract the expert's knowledge is the bottleneck in developing expert systems (22:152; 17:53; 2:228). Knowledge acquisition is a new field and has not evolved to well-defined process (24:43).

Considerations. Knowledge acquisition efforts have to consider the types of knowledge, the domain, and the expert. Recognizing the different types of knowledge helps the knowledge engineer determine the appropriate elicitation technique. A domain expert's knowledge can be classified as explicit or implicit. Explicit knowledge is the knowledge that the expert can articulate. Implicit knowledge poses a more difficult problem, because the expert is not conscious of its presence.

An expert may find it difficult to explain a problem solving strategy based on implicit knowledge. Implicit knowledge takes two forms, knowledge that was once explicit and knowledge from the learning process. Implicit knowledge that was once explicit is that knowledge that experts once learned and consciously applied to problems (1:145). After consciously applying this knowledge again and again, the expert begins to use this knowledge "without thinking." Consequently, the expert loses the ability to verbalize this knowledge (16:154; 1:145). Knowledge from the learning process is even more difficult to acquire.

This form of implicit knowledge has never been expressed explicitly. Rather, the expert has gained this knowledge through experience (1:145). The practical consequence of implicit knowledge is an expert's difficulty in explaining the knowledge underlying problem solving techniques. Currently, no agreed upon, satisfactory method for eliciting implicit knowledge exists (1:149).

The ease of knowledge acquisition is directly related to the problem domain. A domain that does not require the expert system to perform the entire task to be useful facilitates knowledge acquisition. Knowledge acquisition can then focus on one subdomain and expand by including other subdomains. Related to focusing on one subdomain at a time, each subdomain's task should break into subtasks. Each subtask will require a knowledge acquisition effort. Eventually, knowledge of the subtasks will be consolidated for the subdomain task. In turn, knowledge of the subdomains will be consolidated for the entire domain. Also, the domain should be fairly stable. Otherwise, an unstable domain can result in invalidating a substantial part of the knowledge base acquired early. In this situation, the knowledge engineer would have to reaccomplish a major portion of the knowledge acquisition process (24:45).

The knowledge acquisition process relies heavily on the expert's abilities and availability. The expert should have gained expertise by performing the domain task over a long period of time. Furthermore, the experts should have the ability to communicate this expertise to the knowledge engineer in a clear manner (24:44). The most often cited quality desirable in an expert for knowledge acquisition is willingness to cooperate (24:44; 2:230; 15:199). Also, another key consideration is the availability of the expert. The expert should be able to devote a substantial amount of time to knowledge acquisition. Consequently,

strong managerial support is often needed to secure the expert's availability (24:45).

Knowledge Elicitation Techniques. Little or no systematic research has been conducted on the effectiveness of different knowledge elicitation techniques (17:54). Although a direct comparison of techniques is not sensible, a quantitative evaluation of elicitation techniques used by knowledge engineers is needed (2:229). Two broad classes of knowledge elicitation techniques are direct and indirect methods (22:153). Additionally, a third class of techniques exists that uses machine assistance to elicit the knowledge. However, machine assistance techniques are not in wide use (2:229). Recognizing that different types of knowledge are best elicited by different elicitation techniques is an important step in successful knowledge acquisition (2:230).

Direct Methods. Direct methods are used to elicit the expert's explicit knowledge. The most common direct technique is the interview. Interviews can be unstructured or structured. The unstructured interview is effective at obtaining the basic domain knowledge. The more detailed knowledge can be obtained using a structured interview to narrow the focus to gaps in the knowledge (22:153-154; 17:55-56).

A second direct method is to use a questionnaire. Questionnaires are useful in discovering the relationships between objects in a domain and in determining uncertainties tied to the expert's conclusions (22:154).

Discovering how an expert actually solves a problem is the purpose of task observation. In this technique, the expert is allowed to perform without interruption (22:155).

A similar method is protocol analysis, which requires the expert to solve typical domain problems. As the expert solves the problem, the expert is asked to "think out loud" (22:154-155). Variations of protocol analysis use problems with limited information, time or other constraints, or unfamiliar features to yield subdomain and subtle knowledge (17:56-57).

Another direct method is interruption analysis. Interruption analysis allows the expert to perform a task until the knowledge engineer no longer understands the expert's actions. At that point, the knowledge engineer interrupts the expert to ask how and why the expert arrived at that action (22:156).

In the technique of drawing closed curves, the expert indicates relationships among domain objects in a physical space representation. The relationships are indicated by the expert drawing closed curves around the domain objects that are related (22:156).

The final direct method is inferential flow analysis. Beginning with a list of objects from the domain, the expert is asked a series of questions about the relationships between the objects. The result is a network of links among the objects in the domain (22:157). All of the direct methods rely on the expert's ability to articulate the domain knowledge (22:153).

Indirect Methods. Recognizing that not all of the expert's knowledge is explicit leads to indirect methods for knowledge elicitation. Indirect methods are used to elicit implicit knowledge. The different indirect techniques make different assumptions about the expert's underlying representation (physical space, lists, networks, tables, etc.) of the knowledge (22:157-158).

Knowledge with a physical space representation is assumed for the indirect method of multidimensional scaling. In this technique, the expert judges all the possible pairs of objects in a domain for similarity. The judgments are assumed to be symmetric (X is as similar to Y as Y is to X). The result is a representation of objects along user-defined dimensions to reveal clusters of related objects and isolated objects (22:158).

Similar to multidimensional scaling is Johnson hierarchial clustering. However, the key difference is the assumption that the judgments are not assumed to be symmetric. The result is that objects are considered a member of a cluster or not (22:159).

Another indirect method that assumes symmetric judgments is general weighted networks. A network of association, paths among objects, is assumed to be the basis for the judgments. This network reveals *dominating concepts* (objects that have a large number of connections) and members of cycles (objects that are linked in circles) (22:160).

The technique of ordered trees from recall assumes that objects belong to a cluster or not. Also, ordered trees assumes that an expert will recall all the items from a cluster before recalling items from a different cluster. The recalled items in a cluster reflect how the expert organizes knowledge (22:160,162).

Repertory grid analysis incorporates aspects from each of the above indirect methods and is considered the most complete. This technique includes an opening interview followed by analyses on the dimensions and clustering of the domain objects. Basically, this technique is an unstructured session in which the knowledge engineer infers the relationships among the objects in the dimensions that the expert

considers important (22:163). All the indirect techniques attempt to illustrate implicit knowledge (22:166).

Issues in Knowledge Acquisition. Two issues in knowledge acquisition have been the subjects of debate among knowledge engineers. The first issue concerns whether a knowledge engineer should have domain expertise or not. While a general knowledge of the domain is considered necessary, the risk of a knowledge engineer with domain expertise is that the knowledge engineer may bias the knowledge acquisition. The bias may not be limited to problem solving techniques in the domain, but the bias may extend to using only knowledge that meshes with the knowledge engineer's vision of the proposed system (2:229-230). Another argument for knowledge engineers without domain expertise revolves around eliciting implicit knowledge. Knowledge engineers who are experts in the domain could become less sensitive to implicit knowledge (1:149). The arguments for a knowledge engineer with domain expertise center on the fact that the knowledge engineer gains domain knowledge as a result of background research (24:46; 17:55). This knowledge can be used to alert the knowledge engineer to the expert's underlying knowledge representation (17:55). A compromise position is that the knowledge engineer should have the knowledge level of the expected user of the system (2:230).

The second issue concerns the use of a single expert or multiple experts to build the knowledge base. Most expert systems have been built based on a single expert's knowledge (15:199). In complex domains, the problem cited with knowledge acquisition using one expert is that no single expert for the entire domain exists (18:32). Knowledge acquisition with multiple experts offers several advantages. Using multiple experts provides a variety of different kinds of knowledge and

an understanding of the expertise prevalent in the domain (18:33-34). In addition, having more than one expert helps ensure the completeness and accuracy of the knowledge base (17:59). The advantages of multiple experts are accompanied by disadvantages as well. Using multiple experts can result in contradictions in the knowledge base or disagreements on how to approach a problem (15:200; 17:60). An approach to capitalize on the strengths of both approaches is to use one expert to build the initial knowledge base and ask other experts to refine the knowledge (15:200).

Knowledge Representation

The purpose of knowledge representation techniques is to structure the knowledge to make the problem easier to solve (30:392). Essentially, knowledge representation is the way of organizing the knowledge in the knowledge base. New representation methods are the subject of on-going research (12:73). However, the three most common methods of knowledge representation are production rules, frames, and semantic networks (30:63; 12:73).

Rules. Allen Newell is believed to have introduced the concept of production rules to AI (4:6). Expert systems that use production rules for knowledge representation are also called production systems, and the knowledge base is called the rule base (12:74). Production rules, also referred to as rules, are appropriate for representing knowledge of empirical associations developed through extensive experience (30:63). In other words, rules are a good representation method for heuristic knowledge (12:74). Collectively, the rules represent the knowledge in the knowledge base (30:64). Rules are two part statements that contain a piece of knowledge (12:74).

The two parts of a rule are an IF portion and a THEN portion. When the IF portion of a rule is satisfied by the facts or current situation data, the action of the THEN portion is performed (30:64). The comparison of the IF portions of rules to the facts produces inference chains. Inference chains can go forwards or backwards (30:66). In forward chaining, rules are matched to facts to create new facts (30:78). In backward chaining, the system starts with a conclusion to be proved and works backwards through the rules to establish the facts needed to prove the conclusion (30:77).

Since rules represent only a piece of knowledge, a large number of rules may be needed to represent the domain knowledge. Ten to twenty rules is considered a small system, but a viable system usually contains hundreds of rules. Large rule based systems can have thousands of rules (12:75).

The main benefits of rules are ease of maintenance and the ability to handle ambiguity. The structure of the rule base is such that the knowledge is modularized in the rules. This modularity makes adding, deleting, and changing rules relatively quick and easy (12:75). Rules can handle ambiguity through the use of certainty factors and probability (15:43; 12:76-78). A certainty factor attached to a rule indicates the belief, on an arbitrary scale, that the fact or rule is true (15:42; 12:76). Probability is mathematically stronger than certainty factors. In particular, Bayesian probability is used when dealing with inexact data (12:78).

Semantic Networks. One of the oldest and most general representation methods is the semantic network (15:35). A semantic network is a graphical representation of knowledge that illustrates the relationships between objects or concepts in the domain (12:79). In the

network, objects and concepts are represented by nodes and linked by arcs. The arcs describe the relationships between the nodes (12:80; 30:70; 15:35). Common arcs are IS-A and HAS-A relationships between nodes (30:70; 15:35). Other types of arcs are definitional and heuristic. The semantic network structure provides flexibility in adding new nodes and links (15:36). An important feature of the semantic network is inheritance. Inheritance is the ability to inherit a characteristic from a related node (30:71; 15:36; 12:81). This ability reduces redundancy, but makes handling exceptions difficult (15:37).

Frames. Another method for representing knowledge is through the use of frames. Frame systems are appropriate for domains where the form and content of the data are critical to problem solving (30:74-75). Frames are analogous to nodes in semantic networks (30:74; 12:82). However, frames have the added dimension of slots that store information associated with the object or concept (30:74; 15:44; 12:82). Slots may contain values, pointers to other frames, or sets of rules (15:44). Also, each slot can have a procedure attached so that if the value in the slot is added, deleted, changed, or absent, the procedure executes (30:74; 15:44; 12:82). Continuing the analogy, frames may be linked together to allow for inheritance. Interconnected frames provide for a very detailed knowledge base of rich information (15:44,46; 12:83).

Expert System Development Tools

Expert system tools are programs or a collection of programs to simplify development of an expert system (30:80; 12:113). Development tools can be thought of as existing on a continuum. Programming languages are on the far left on the continuum, and shells are on the far right of the continuum (15:83; 12:113). Each broad class of tools,

programming languages and shells, has its own advantages and disadvantages (30:80). Therefore, selection of a development tool can enhance or hinder the development of the expert system (15:82).

Programming Languages. A programming language is a means of communicating with a computer to control and direct its operation (30:80). Programming languages, for the purposes of AI, are broken into two categories, problem oriented and symbol manipulation languages (30:80; 12:114-115). By definition, problem oriented languages are designed to solve a particular class of problems (30:80). Expert systems have been developed using problem oriented languages such as FORTRAN, PASCAL, C, and BASIC (12:114). The major drawbacks with using problem oriented languages are computing speed and development time. For computing speed, the trend in AI is towards C, which runs faster than most other problem oriented languages. However, expert systems generally run slow when programmed in a problem oriented language because of the complex search and pattern recognition activities. The other drawback is the time required to develop an expert system in a problem oriented language. The entire system must be developed from scratch, from the structure of the knowledge base to the inference engine to the user interface. Programming every aspect of an expert system is a challenging task even for a skilled programmer (12:114-115).

Symbol manipulation languages are designed for AI applications (30:81). Designed to handle symbolic processing, symbol manipulation languages make developing an expert system much easier than problem oriented languages (15:83). The two most popular symbol manipulation languages are LISP and PROLOG (12:115). LISP owes its dominance and longevity to such features as "easy and flexible symbol manipulation, automatic memory management, sophisticated editing and debugging aids,

and the uniform treatment of program code and data" (30:82). Since its development in the 1950s, LISP has spawned many different "dialects" (12:115; 30:82; 15:85). The many different dialects allow LISP to be run on a variety of hardware, but the code is not always portable between hardware configurations (12:115-116). While LISP has been the dominant language in the United States, PROLOG has gained wide acceptance in the international community and growing use in the United States (15:87; 12:129). PROLOG is based on controlled logical deduction (15:88). Programming in PROLOG uses facts and rules about objects and the relationships between objects to answer user queries (12:126; 15:88). The main difference between LISP and PROLOG is in the approach to the program's task. LISP outlines the steps to be taken to accomplish the task. In contrast, PROLOG describes the task within a set of constraints to be satisfied. Simply, LISP specifies the "how" of a task, and PROLOG specifies the "what" of a task (15:88). PROLOG and LISP are the dominate symbol manipulation languages; however, other languages, such as POPLOG, are in use.

The main advantage of symbol manipulation and problem oriented languages is the flexibility the programmer has in representing the knowledge and in building the mechanisms to manipulate the knowledge. On the other hand, the high degree of flexibility comes at the expense of any guidance with respect to knowledge representation or inference engines (30:82-83). This guidance is available, at the expense of flexibility, in expert system shells.

Expert System Shells. A shell is a collection of programs to develop an expert system without using a programming language (12:130). A shell has all of the necessary components of an expert system, but the knowledge base is empty (5:30-31; 12:130)). Consequently, shells have

two advantages in developing an expert system. The first advantage is that a shell allows rapid development, since only the knowledge base has to be created (12:131; 15:81). The second advantage is the specific knowledge representation, inference, and control schemes that help model the problem or task (5:31; 15:81). The disadvantages of shells are the lack of flexibility and the limitation of a shell to a particular class of problems (30:83). Two classes of shells are rule-based and induction. Different rule-based shells may have different inference engines or user interfaces, but all rule-based shells allow the developer to enter production rules for representing the knowledge. Induction shells use examples from the problem domain. An example is a list of problem attributes that result in some decision or outcome. The attributes and decisions or outcomes of the examples are entered into a matrix. The shell uses the matrix to generate rules (12:131,135).

Selecting a Tool. Choosing the right tool for building an expert system is one of the most difficult decisions to make. For any problem, there may be more than one tool that could perform adequately. However, there may be no tool that is perfect for that task (30:142). The selection of the tool is a two step process. The first step is to initially select the tool, and the second is to evaluate the tool (30:143).

In selecting a tool, the tool's features should match the characteristics of the domain, of the likely solution, and of the system itself. The domain characteristics include the form of the data, the size of the search space, and the problem structure. Pertinent characteristics of the likely solution are the representation and search techniques used in the system. The system characteristics include the type of user interface and the method for modifying the system

(30:146,148; 16:211). Additional considerations in selecting a tool are the resource constraints (time, money, personnel, and hardware), desired support facilities (debugging aids, explanation subsystems, etc.), maturity of the tool, and maintenance of the tool. The resource constraints primarily influence the selection of the type of tool, programming language or shell (30:143-144). While the resource constraints exist a priori, the complex set of tool requirements is derived from the uses of the system. The system's uses depend on who the user is and at what stage in development the system is (5:38). After selecting the tool, the next step is to evaluate the tool.

In evaluating a tool, the tool's intended use must be considered. Criteria for evaluating tools are the tool's basic features, development environment, functionality, support, and cost (27:167). The tool's basic features, such as the knowledge representation scheme and inference formalisms, should be appropriate to the problem (27:168; 11:71-72). The development environment should have the support facilities required by the developer at each stage of development (27:169; 11:70)). The functionality of the tool refers to how the tool actually works, not just a description of the tool's features. For example, some functionality issues are ease of use, robustness, capability to run on a variety of hardware, and limits on the number of rules or frames (27:169; 11:73). Support for a tool is critical to the inexperienced expert system builder. The vendor of the tool should provide training, documentation, and technical support to the builder (27:169; 11:73). The last criterion is cost. Using cost to evaluate a tool focuses on the cost of the tool, training, technical support, and any required hardware (27:170; 11:73). An approach for using these criteria is to build a

small prototype system using a representative problem to evaluate the adequacy of the tool (30:149).

Expert System Development Methodology

Building an expert system is not a well-defined sequence of steps. Therefore, many system builders have adopted an evolutionary approach to developing expert systems (30:135). Several apparently different development methodologies exist among expert system builders. No two methods are identical in the exact name, number, and sequence of steps in the development process, but each method shares common aspects with the other methods. The common aspects can be viewed as the interdependent and overlapping stages of identification, conceptualization, formalization, implementation, and testing (30:136; 16:140).

Prior to identification, the knowledge engineer must determine if the domain is suitable for an expert system application. This determination is a critical step in the development process (25:26; 12:150). During identification, the knowledge engineer identifies the participants, problem characteristics, resources, and goals iteratively. Before beginning any knowledge acquisition, the knowledge engineer must select an expert and define the roles of any other participants (30:136; 16:141). Next, the knowledge engineer and the expert identify the type, scope, and structure of the problem (15:180; 12:150; 30:136; 16:141). The problem characteristics will suggest needed resources for development. The goals of the expert system under development are identified during this stage, also (30:136; 16:140-143).

The second stage is conceptualization. Conceptualization focuses on the key concepts and relations of the problem in an iterative manner. Examples of key concepts and relations are subproblems, domain object

relationships, and problem solving strategies. An important point in conceptualization is not to attempt to do a complete problem analysis prior to initially implementing a system. The initial implementation will provide information that will influence conceptualization (30:137; 16:143-144).

Formalization uses a framework to represent the key concepts and relations of the problem. The framework is usually provided by some development tool. At this point, the knowledge engineer must determine which tool best matches the problem characteristics (30:137-138; 16:146; 15:178; 12:159). The result of formalization is a body of formalized knowledge ready for implementation (30:138).

During implementation, the formalized knowledge is used to create a program (30:138; 16:146; 12:168). The knowledge specifies the contents of the data structures and the inference and control strategies. The development tool specifies the program's form. Program integration is achieved by ensuring consistency in the knowledge and inference mechanisms (30:138; 16:146). The program created during this stage becomes the system prototype (30:138; 16:146; 12:168; 15:191).

The final stage, testing, involves evaluating the system's performance and revising the prototype as necessary (30:138; 16:147-148; 12:169; 15:193). Evaluating the prototype's performance centers on the system's ability to solve the problem in an efficient and effective manner. The efficiency of the system relates to the technical aspects of the prototype. The effectiveness of the system relates to the ability of the system to help the user in a significant and timely manner. After testing, the prototype is revised to correct any deficiencies in the system (30:138-139; 16:147-148).

Expert System Technology and Technical Order Acquisition

The previous sections discussed expert system technology devoid of any particular problem domain. Having evolved from laboratory experiments to practical applications, expert systems have been classified according to the type of problem domain addressed by the system (15:2-3; 30:33). Expert systems have found applications in a diverse group of problem domains (30:40). Each problem domain has a body of knowledge that is needed to work in that domain. The ability to represent that body of knowledge is the key characteristic of an expert system (30:24; 13:34). Consequently, knowledge is the heart of the structure of an expert system (30:16). Acquiring and representing the knowledge are the responsibility of the knowledge engineer. In addition, the knowledge engineer selects a development tool to aid in building the expert system. Building an expert system is not constrained to a single accepted development methodology; however, most methodologies share common aspects (30:135-136; 16:140). In addressing a particular problem domain, expert system technology moves from theory to practice.

In order to apply expert system technology to TO acquisition, the process and problems of acquiring TOs must be understood. The next two sections covers the basic process in planning and executing a TO acquisition program. In addition, the problems in TO acquisition that could be eased by expert systems technology are discussed.

Technical Order Acquisition Process

TOs describe the procedures for operating and maintaining Air Force systems and equipment (8:4). Indeed, Air Force policy states that systems will not be fielded without verified TOs (8:2). Consequently, the importance of a well managed TO acquisition program is evident from a

readiness standpoint. From a financial standpoint, the Department of Defense spends millions of dollars each year acquiring TOs. At a cost of $600 to $1200 per page, TOs represent an investment in the supportability of any weapon system (20:25). The investment in TO acquisition is made as a part of the weapon system acquisition. The acquisition of TOs is not without its problems (3:2).

TO acquisition is guided by AFR 8-2 Air Force Technical Order System and TO 00-5-1 AF Technical Order System. These two documents provide the basis for TO acquisition efforts. The TO acquisition cycle can be viewed as a series of steps. The steps include the TO acquisition planning, contractual documentation, TO guidance conference, TO program progress monitoring, in-process reviews (IPRs), validation, verification, and pre-publication review. These steps occur within the overall framework of the weapon system acquisition cycle (8:7-9; 9:3-1 to 3-5).

Technical Order Acquisition Planning. The early stages of a weapon system acquisition allow for planning the acquisition of the needed TOs. The Statement of Need (SON) provides the initial description of the equipment and the maintenance concept. This description is used to formulate the TO acquisition strategy (19:24). The strategy is documented in the Technical Order Development Management Plan, which is written and updated by the Technical Order Management Agency (TOMA, office responsible for acquiring the TOs) (8:4). Also, the early stages are marked by evaluating different technologies and alternatives for increasing readiness and decreasing costs in developing TOs (19:24). Subsequently, the TO planning must be manifested into contractual documents.

Contractual Documentation. Once the plan for TO acquisition has been developed, the TOMA must ensure that the Statement of Work (SOW),

Contract Data Requirements List (CDRL), and the Technical Manual Contract Requirements (TMCR) will culminate in the needed TOs. The SOW tasks direct the contractor to perform TO related tasks such as integrating Logistics Support Analysis data into TO development (9:3-1). The performance of the SOW tasks results in documentation which is delivered in accordance with the CDRL. Typical CDRL entries for TO development include Technical Manual Schedules and Status Report and Technical Manual Publication Plan (19:25). The TMCR is a result of a 1985 memorandum from the undersecretary of defense for research and engineering (20:27). The TMCR gives the TOMA and the contractor a contractually binding document that covers the general and specific requirements for TO development, the deliverable technical manuals, and applicable specifications and standards (TMCR:2). Each of the contractual documents must be tailored by the TOMA to accomplish the TO development at the lowest possible cost (19:27).

Technical Order Guidance Conference. Following the awarding of the contract, a TO Guidance Conference is conducted. The purpose of the guidance conference is to ensure mutual understanding of the contractual requirements. The participants in the conference include the using, supporting, and acquiring command as well as the contractor. Basically, the conference gives the contractor the opportunity to present the contractor's interpretation of the SOW, CDRL, and TMCR. Any differences in interpretation between the Air Force and the contractor are referred to the contracting officer for resolution (9:3-2).

Technical Order Program Monitoring. The TOMA monitors the contractor's progress in developing the TOs through a CDRL delivery, the Technical Manual Schedules and Status Report. This report provides

information on the schedule for TO development and the status of each manual (19:25).

<u>In-Process Reviews</u>. IPRs are conducted by the TOMA to ensure that the contractor is preparing the TOs in compliance with the contractual requirements. The timing and number of IPRs for a weapon system or equipment is negotiated on an individual program basis. The focus of the IPRs is on evaluating the format and technical content of the TOs. The using, supporting, and training commands participate in the IPRs (9:3-2).

<u>Validation</u>. Validation is the testing of a TO for technical accuracy by the contractor's personnel. Validation is usually conducted at the contractor's facility, but validation can be conducted at an operational site if specified. The TOMA, or a designated representative, witnesses the contractor's validation efforts to attest to the performance of the validation. The actual validation entails the contractor's personnel using the TOs to operate and maintain the equipment. The contractor must use one of three methods for validation: demonstration, simulation, or desk-top analysis. The method chosen by the contractor is subject to Air Force approval. Following validation, the contractor incorporates any changes identified during the validation prior to verification of the TOs (9:3-3 to 3-4).

<u>Verification</u>. After the contractor has validated the TOs, the Air Force uses its own personnel to test and prove the adequacy of the TOs for equipment operation and maintenance. This process is called verification. Verification consists of Air Force personnel using the validated TOs to operate and maintain the equipment. The personnel should be of the same specialty code and skill level as those who are projected to operate and maintain the equipment when fielded.

Verification usually starts during testing so that the TOs will be verified in sufficient time to print and distribute the formal TOs. The main result of verification is the certification that the TOs are technically accurate and match the hardware configuration. Following verification, the contractor corrects any deficiencies in the TOs identified during verification (9:3-4 to 3-5).

Pre-Publication Review. The pre-publication review is the final review of the TOs before the contractor prepares the TO reproduction media. The purpose of this review is to ensure that all the verification comments have been incorporated and that the TOs meet the contractual requirements for format. A successful pre-publication review permits the contractor to prepare the reproducible media for printing and distribution of the TOs by the responsible Air Logistics Center.

Technical Order Acquisition Problems

The TO acquisition process is not without some persistent problems. In their 1984 Master's thesis, Brown and Lyon highlighted four problem areas of TO acquisition. The problem areas were lack of early planning for TOs, poor communication and coordination between all agencies involved in the process, inadequate manpower, and inadequate training and assistance for TO acquisition personnel (3:4). The aforementioned memorandum that established the TMCR has raised the visibility and increased the standardization of TO acquisition (20:26). Consequently, the TMCR has directly and indirectly helped ease these four problems areas. However, the problems of inadequate manpower and training continue.

Inadequate Manpower. The research conducted by Brown and Lyon indicated that 74% of the 130 interviewees believed that the manpower for

TO acquisition was inadequate (3:61). Assignment of TO personnel was not based on any standard or criteria (3:23). Some small programs do not have a dedicated TO manager. Rather, the DPML or ILSM must oversee several logistics areas simultaneously (20:26). Brown and Lyon recommended that a TO management center be established to assist the smaller programs in TO acquisition. The result would be a reduction in the overall amount of manpower required (3:33-34). The manpower problem is closely related to the problem of inadequate training for TO acquisition personnel (3:67).

Inadequate Training. Brown and Lyon concluded that the largest single problem with the TO acquisition process is the lack of training for TO acquisition managers. Training included experience and corporate memory as well as formal education (3:67). An interview with the chairperson of the Centralized Technical Order Management (CTOM, the group that manages the Air Force TO system) Executive Committee elicited the following response.

> Corporate memory in the technical order acquisition process is very low. There is no formal career field for technical order acquisition managers. Personnel performing duties as technical order acquisition managers usually have little or no experience in the field prior to being assigned to a specific project. They learn through trial and error, then they are transferred and their knowledge goes with them [3:23].

The CTOM members and the TO managers interviewed by Brown and Lyon emphasized the need for individual assistance from an "assistance agency" (3:25). Mr. Munguia, the course director for the Air Force Institute of Technology Technical Order Acquisition Management Course, stated in his interview that TO acquisition managers "need experts and specialists to turn to for assistance" (3:26). One of the recommendations to alleviate the problem of inadequate training offered by Brown and Lyon is a TO management center that would provide "expertise and assistance" (3:85).

Summary

This chapter reviewed some of the current literature on expert systems and TO acquisition. The first three sections of this chapter pertained to the evolution, types, and applications of expert systems. The next two sections covered the characteristics and structure of expert systems. The last four sections on expert systems addressed the practical knowledge required for developing expert systems. This practical knowledge was divided into knowledge acquisition, knowledge representation, development tools, and development methodology. The two sections on TO acquisition related to the TO acquisition process and the process's problems. The next chapter discusses the research methodology.

III. Methodology

Overview

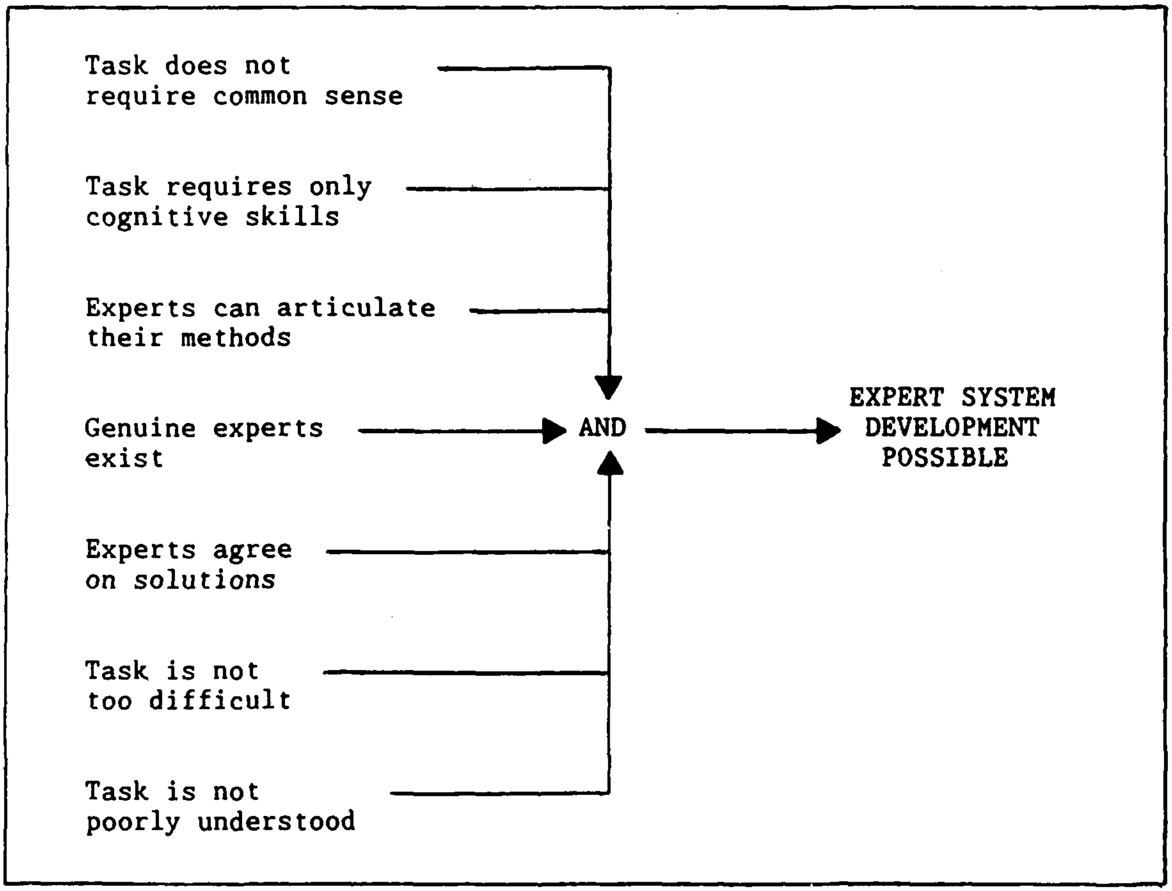
This chapter discusses the methodology that was used to develop the prototype expert system for one ILS element. The first step in the methodology focused on selecting the problem domain (the ILS element) for the expert system application. The second through sixth steps in the methodology followed the phases of expert system development methodology discussed in the literature review. In order, these steps were identification, conceptualization, formalization, implementation, and testing phases.

Selection of Problem Domain

The selection of the problem domain was made using Waterman's criteria to determine if expert system development was possible, justified, and appropriate (30:128-133). First, the characteristics of the domain were examined to support the contention that expert system development is possible. Next, the criteria for justifying expert system development was applied to the domain. Finally, the domain was compared to the criteria for the appropriateness of expert system development.

The examination of the problem domain characteristics focused on whether the domain meets the criteria to determine if an expert system is possible. For expert system development to be possible, the domain had to successfully meet each criterion. The criteria is summarized in Figure 2. The problem domain should not require common sense, in that common sense is a broad area of knowledge rather than specific expertise. No physical skills should be required, because only the expert's thought and reasoning processes can be represented by the expert system.

Experts who can successfully articulate their approaches to problem solving are absolutely essential. Obviously, recognized experts must exist and must agree on the solution to the problem in order to build an expert system. Finally, the problem and the problem solving techniques must be well understood, and the problem must be narrow enough not to require a large amount of knowledge from different areas (30:128-129).

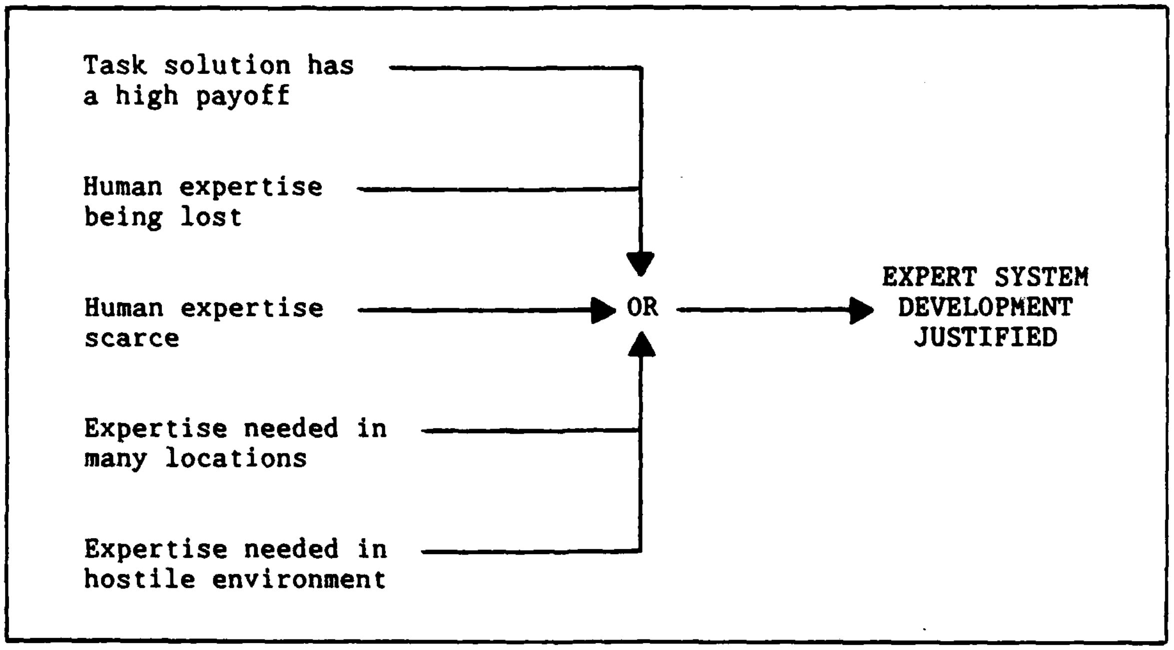


(30:129)

Figure 2. Requirements for Expert System Development

Since developing an expert system is a complex and time consuming task, the criteria for justifying expert system development was applied to the domain. For expert system development to be justified, the domain had to meet at least one of five criterion. The criteria are

summarized in Figure 3. First, an expert system can be justified if there is a large cost savings or productivity increase possible. Second, if the expertise is being lost (through retirement or transfers) then an expert system is justified. Third, scarce and, consequently, expensive expertise is justification for expert system development. Fourth, many different people in many different locations needing the same expertise can justify an expert system. Lastly, an expert system is justified if the expertise is needed in an environment too dangerous for human experts (30:130-131).

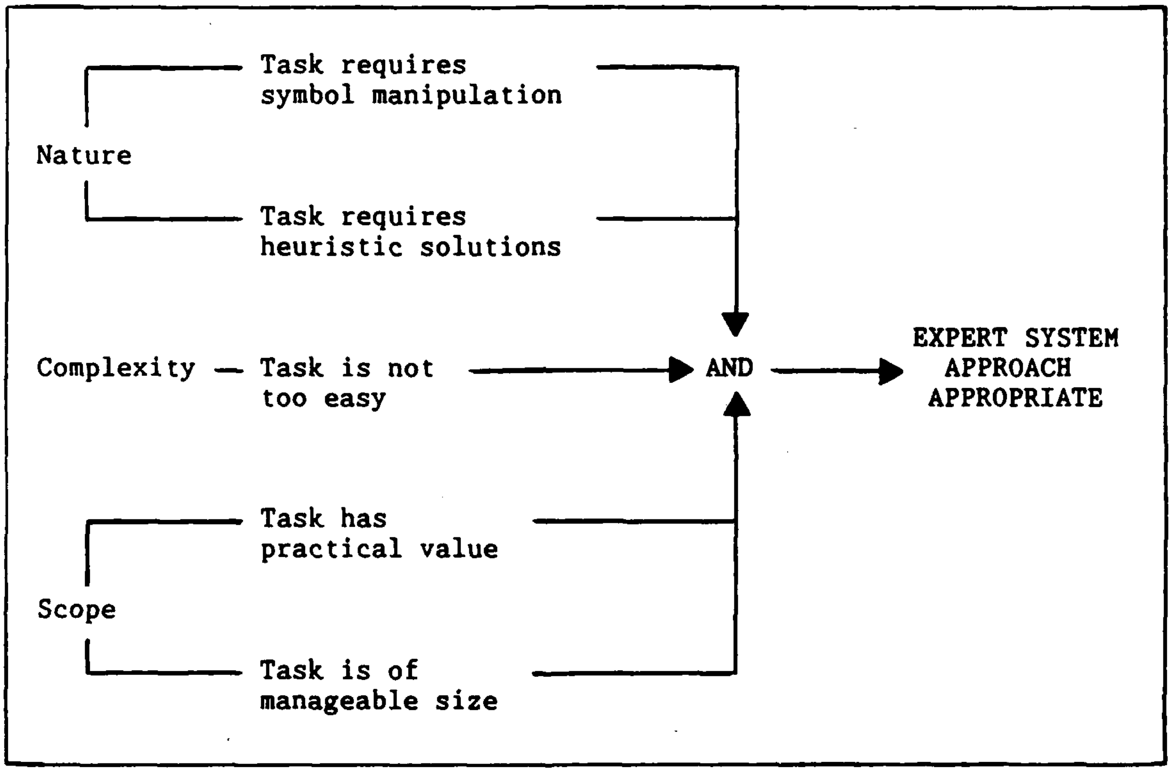


(30:130)

Figure 3. Justification for Expert System Development

Determining the appropriateness of using an expert system for the domain centered on the problem's nature, complexity, and scope. For expert system development to be appropriate, the domain must meet each and every criterion for appropriateness. The criteria are summarized in Figure 4. The nature of the problem should be that it can be

represented using symbols and solved through the use of heuristics (rules of thumb). The problem's solution should be the culmination of the expert's years of experience. Otherwise, the problem may be too easy to solve, and the time and effort to build an expert system are not warranted. The problem should be of manageable size and have a solution with practical value (30:131-134).



(30:132)

Figure 4. Characteristics That Make the Use of Expert Systems Appropriate

Having determined whether expert system development is possible, justified, and appropriate for the domain, the focus of the research moved to the first phase of the prototype development methodology.

Identification Phase

After determining that the domain was suitable for an expert system application, the first phase in the development process was

identification. During identification, the knowledge engineer identified the participants, problem characteristics, resources, and goals for the prototype system. Before beginning any knowledge acquisition, the knowledge engineer selected an expert and defined the roles of the other participants (30:136; 16:141).

The expert was selected based on four factors. The first factor was that the expert have gained domain expertise through performing in the problem domain over a long period of time. Second, the expert must have the ability to clearly communicate the expertise to the knowledge engineer. The third, and perhaps the most important, factor was the expert's willingness to cooperate during the prototype development. The final factor considered in selecting the expert was the amount of time the expert was able to devote to knowledge acquisition and evaluation (24:44-45). After selecting the expert and securing the expert's support, the knowledge engineer gave the expert an introductory lesson on expert systems and guided the expert through an example system. The roles of other participants in the development were identified by the knowledge engineer. Primarily, these participants, other domain experts and intended users, were selected to serve as evaluators during the prototype evaluation.

Next, the knowledge engineer and the expert identified the characteristics and scope of the problem (15:180; 12:150; 30:136; 16:141). Prior to meeting with the expert, the knowledge engineer reviewed literature pertinent to the domain that was suggested by the expert. Identifying the problem characteristics was achieved through an exchange of ideas and views between the expert and the knowledge engineer. This exchange occurred as the expert worked through a typical domain problem. As the expert worked, the knowledge engineer took

careful notes and interrupted only when the engineer did not understand the expert's actions or reasoning. The expert addressed or was asked to address the following problem characteristics: characterization of the problem, important subproblems, partitioning of tasks, required data, domain terminology, and attributes of an acceptable solution. These problem characteristics were used by the expert and knowledge engineer to arrive at the key concepts in the domain and scope the problem (16:141-142). Determining the scope of the problem was accomplished by a subjective assessment by the expert. The expert's assessment identified the areas of the domain that would be manageable in size and of practical value for expert system application (30:136-137). Consideration of the scope of the problem and the problem characteristics aided in identifying the resources required for developing the prototype.

Identifying the resources required to develop the prototype centered on sources of knowledge, time, and computing facilities. The sources of domain knowledge identified by the expert as relevant to the development efforts included reference documents, examples of problems and solutions, and past experience. Relevant knowledge identified by the knowledge engineer included knowledge acquisition techniques, knowledge representation schemes, and expert system development tools (16:142). In addition, the knowledge engineer's background included limited experience in the domain. The time required of the domain expert for knowledge acquisition and prototype evaluation was estimated by the knowledge engineer and agreed to by the expert. Also, the knowledge engineer estimated the time required for developing and evaluating the system after knowledge acquisition. The final resource identified was the needed computing facilities. The computing facilities were determined by the knowledge engineer. Since the Air Force standard for microcomputers

is the Zenith Z-248, the hardware was constrained to the Z-248 and compatibles. The software requirements were determined to be an expert system development shell because of the time constraints on the research. Following identification of the necessary resources, attention turned to the goals of the prototype system.

In addition, the goals of the prototype system developed were identified by the domain expert and the knowledge engineer during this stage (30:136; 16:140-143). In determining the problem characteristics and the scope of the problem, the expert and the knowledge engineer discussed what the goals of the prototype would be. The goals were used to assist in estimating the merits of different problem approaches. In turn, the goals directed the desired output of the prototype. Lastly, the knowledge engineer considered any possible external constraints that could impact the system.

Conceptualization Phase

The second phase was conceptualization. Conceptualization focused on the key concepts and relations of the problem discovered in the identification phase. The activities of this phase were primarily related to elucidating these concepts and relations by acquiring the knowledge from the expert in an iterative manner (30:137; 16:143). Because of the expert's demonstrated ability to communicate the expertise effectively during the identification phase, the knowledge engineer elected to use a direct method for eliciting the expert's knowledge. The method used by the knowledge engineer was unstructured interviewing.

The knowledge engineer conducted a series of seven unstructured interviews with the expert. Each interview lasted approximately ninety minutes. All the interviews were scheduled at the convenience of the

expert over a period of two weeks. Additionally, the expert was receptive to answering any specific questions that occurred to the knowledge engineer by telephone. Of the seven interviews, five were conducted at the expert's desk and the remaining two in nearby conference rooms.

The knowledge engineer interviewed the expert for the purpose of expanding and building on the key concepts and relations. Therefore, the interviews addressed the available data, subtasks, domain object relationships, problem solving strategies, and justification and explanations for actions (16:143-144). During the course of the interviews, the expert would discuss a key concept starting with a general approach sometimes using reference documents. As the discussion on the concept became more thorough, the expert provided examples of problems and solutions pertinent to the concept. Also, the expert drew on past experience to relate ideas about the concept to the knowledge engineer.

Using the notes from the interviews, the knowledge engineer wrote down the key concepts and relations. At this point, the knowledge engineer began to consider which representation schemes and shells might be suitable for formalizing the knowledge gained from the expert. The next phase is formalization (30:137; 16:143-144).

Formalization Phase

The formalization phase involved choosing a representational scheme for the key concepts and relations of the problem. In turn, the chosen knowledge representation scheme influenced the selection of the development tool. During this phase, the knowledge engineer's role became more active as decisions about the technical aspects of the

prototype had to be made (16:144). The goal of the formalization stage was to arrive at a body of knowledge ready to be implemented using the development tool (30:138).

The knowledge engineer examined the domain knowledge acquired from the expert and the problem characteristics identified during the conceptualization and identification phases, respectively. This examination was made to ascertain which knowledge representation scheme (rules, semantic network, or frames) is most suitable. The knowledge engineer concentrated on matching the scheme to the most important concepts and minimizing the representational mismatch of the remaining domain knowledge (16:146).

In assessing the applicability of rules as the representation scheme, the knowledge engineer focused on determining if the expert's knowledge was a result of observed relationships through years of working in the domain (30:63). Another aspect of using rules that warranted investigation was the issue of forward or backward chaining inference techniques. In backward chaining, the system begins with a goal of proving "X" and executes only the rules needed to prove "X" (30:68). In contrast, a forward chaining system matches rules against facts about the current situation to establish new facts (30:67). The importance of the distinction in chaining techniques is that a system with the goal of inferring one particular fact is best suited for backward chaining (30:67).

Continuing the search for a representation scheme, the knowledge engineer scrutinized the appropriateness of a semantic net. The requirement for the strengths of semantic networks were assessed by the knowledge engineer. The knowledge engineer judged the importance of a

semantic network's ability to represent links between concepts, to inherit properties, and to classify the domain knowledge (30:70-72).

The final representation scheme considered was the use of frames. Frames are used extensively in problem domains where the form and content of the data is crucial to problem solving (30:75). For this reason, the knowledge engineer appraised the criticality of the form and content of the data in the expert's problem solving strategies (30:74-75). Since a frame-based system has much of the basic qualities of the semantic network, the knowledge engineer did not reassess the importance of representing the links between concepts. Rather, the knowledge engineer focused on the possible advantages or disadvantages of being able to describe features of each concept through slots (30:74).

After deciding on the knowledge representation scheme, the knowledge engineer selected an expert system shell to aid in developing the prototype. Initially, programming languages were considered as a possible tool for development; however, the knowledge engineer's lack of experience and the time constraint eliminated programming languages as a viable tool. Instead, the knowledge engineer chose to use a shell for the guidance with respect to knowledge representation and inference mechanisms. The shell was chosen on the basis of several criteria. Foremost, the shell had to match the problem domain characteristics and support the selected representation scheme (11:70-71). Other desirable traits included a short (three days or less) training period to learn how to use the shell and an extensive on-line help feature (11:72). Additional features considered important were the ability to accommodate uncertainty factors, ease of user interface, and cost (low cost would facilitate the use of the prototype in the field) (11:71-72). The last feature of the shell was considered a "must have" - compatibility with

the Zenith Z-248. Having chosen the representation scheme and the shell, the next stage was to implement the knowledge with the aid of the shell.

Implementation Phase

During implementation, the formalized knowledge was used to create a program (30:138). The program code created during this stage became the system prototype. The prototype was a result of the knowledge engineer using the acquired domain knowledge to create the knowledge base for the system. Also, the domain knowledge was used to determine the inference and control strategy. The actual form of the program was specified by the shell selected during the formalization phase. The shell served to ensure consistency in the knowledge base and inference mechanisms of the prototype (30:138).

In creating the knowledge base, the knowledge engineer referred to the knowledge acquired from the expert and the pertinent literature. This knowledge was the basis for determining the requirements for variables (16:146). The contents of each variable was then defined along two dimensions. The first dimension was whether the contents of the variable were given or inferred. If given, the second dimension concerned whether the user would select the contents from a system provided list or input the contents arbitrarily. In addition, particular attention was paid to the order that the variables were prompted. The order was important because prompting the user in a natural way can increase the user's confidence in the system (30:138).

Another consideration in the implementation phase was the inference and control strategy. The choice of the inference and control strategy of the prototype was limited to the inference mechanisms provided by the shell. Using the information flow that emerged from the knowledge

acquisition, the knowledge engineer attempted to match, as closely as possible, the information flow to one of the shell's inference mechanisms. More specifically, the selection of the inference mechanism revolved around the order in which facts from the knowledge base are applied and how those facts are used to draw inferences (30:22-23; 16:18).

The form of the program was dependent on the shell (30:138). The knowledge engineer spent approximately three days reading the shell's reference manual and using the example systems. Consequently, the knowledge engineer learned how to use the shell to build the knowledge base for the prototype. The manual specified such shell conventions as maximum length of variable names, allowable special characters, keywords, and comment lines (VPEXPERT:4.3). These conventions and other formatting and syntax rules governed the form of the program. The last part of the implementation phase was to write the code for the knowledge base and check the code for consistency using the shell's consistency checker. With the prototype implemented, the next phase was to test the prototype.

Testing Phase

The final stage, testing, involved evaluating the system's performance and utility (30:138; 16:147-148; 12:169; 15:193). With this prototype system and other expert systems, evaluation was difficult because there was no formal way to prove an answer was correct or the best possible choice (30:198). One reason for this difficulty is that the domain experts are not evaluated objectively. Even with the abundance of literature on testing humans, the methods and results do not seem to apply to testing expert systems or the human counterparts

(16:242-243). As a result, in some domains it is difficult to decide what qualifies one as an expert in the domain (16:264). At this stage in expert system technology, the evaluation process is more an art than a science (16:277).

Despite the lack of consensus in the field of expert systems on how (or when or why) to evaluate expert systems, a two part testing program was devised (16:243). The first part was aimed at system examination and refinement by the domain expert. The second part was aimed at validating the system using additional experts and potential end-users (30:160). The purpose of the testing phase was to test the prototype's competence in the domain and determine whether the system produces meaningful results (16:245).

Part one of the testing program was conducted by the knowledge engineer and the domain expert. This testing was performed for the knowledge engineer and the expert to look for deficiencies in the prototype. The expert's role in this part of the testing was to assess the accuracy of the system's knowledge and performance. To perform this assessment, the expert evaluated the adequacy of the system's output and the correctness of the system's reasoning processes (16:255). The evaluation was accomplished by having the expert examine and critique the knowledge base. In addition, the expert compared the prototype's control strategies with the methods used by the experts (30:160). Next, the expert provided two cases, previously solved by the expert, for the system to solve. The system's output was compared to the expert's solution to highlight any discrepancies between the two solutions (30:160). Regarding performance, specific questions asked of the expert at the end of this part of the testing were as follows:

1. Was the system's output (advice and decisions) appropriate?

2. Is the knowledge base correct, consistent, and complete?

3. Do the control strategies consider items in the natural order?

4. Are the explanations of the "how" and "why" of actions and conclusions adequate [30:138]?

With a system that satisfied the expert, the prototype moved to the second part of testing.

The second part of the prototype testing focused on validating the system. The testing was conducted by the knowledge engineer, additional experts, and potential end-users. First, the knowledge engineer presented the cases from part one of the testing to two additional experts. The additional experts were asked to assess the adequacy of the system's output (30:160). Next, the additional experts provided a case for the system to solve. Once again, the additional experts were asked to assess the system's output as compared to their own solutions (30:175). The following questions concerning performance were asked of the additional experts.

1. Was the system's output (advice and decisions) appropriate?

2. Do the control strategies consider items in the natural order?

3. Are the explanations of the "how" and "why" of actions and conclusions adequate?

4. What changes to the system would you recommend [30:138]?

The potential end-users evaluated the utility of the system. Using the cases from part one of the testing, the end-users used the prototype to evaluate the usefulness of the system's output, ease of use, and speed (16:244). The following questions concerning the utility of the system were asked of the end-users.

1. Is the system output helpful in a meaningful and significant way?

2. Is the output organized well and presented at the right level of detail?

3. Does the system's response time seem excessive?

4. Is the system interface friendly enough?

5. What changes to the system would you recommend [30:138]?

Following the evaluation by the additional experts and the potential end-users, the knowledge engineer recorded the recommended changes to the prototype. Of course, the ultimate test of an expert system is whether the system is actually used by anyone other than the developers (16:245).

Summary

This chapter discussed the methodology that was used to develop the prototype expert system for one ILS element. The first section of the chapter focused on selecting the problem domain for the expert system application. The second through sixth sections of the chapter addressed the phases of expert system development methodology discussed in the literature review. In order, these phases were identification, conceptualization, formalization, implementation, and testing. The next chapter discusses the results of this research.

IV. Analysis and Results

Overview

This chapter analyzes and records the results of the research conducted in developing the prototype system. Each step in the development methodology (selection of the problem domain, identification, conceptualization, formalization, implementation, and testing) yielded results that are discussed in this chapter. Consequently, the sections of this chapter parallel the steps in the methodology as presented in chapter three.

Results of the Problem Domain Selection

The results of the first step, selection of the problem domain, centered on Waterman's criteria for expert system development (see Figures 2, 3, and 4). The knowledge engineer selected TOs as the candidate problem domain. Following the initial selection of TOs, the knowledge engineer evaluated the problem domain of TOs against the criteria offered by Waterman. The results of examining the problem characteristics to determine if expert system development was possible are shown in Table 1. Since the problem domain of TOs met each and every criterion, expert system development for TOs was considered possible. The results of examining the problem domain to determine if expert system development was justified are shown in Table 2. Since the problem domain of TOs met at least one of the criterion, expert system development for TOs was considered justified. The results of examining the problem domain to determine if expert system approach was appropriate are shown in Table 3. Since the problem domain of TOs met

all of the criteria, an expert system approach to TOs was considered appropriate. From this first step in the methodology, the knowledge engineer determined that expert system development for TOs was possible, justified, and appropriate.

Table 1

Requirements for Expert System Development Applied to TOs

Criteria	Criteria Met by Problem Domain	
	Yes	No
Task does not require common sense	*	
Task requires only cognitive skills	*	
Experts can articulate their methods	*	
Genuine experts exist	*	
Experts agree on solutions	*	
Task is not too difficult	*	
Task is not poorly understood	*	

Table 2

Justification for Expert System Development Applied to TOs

Criteria	Criteria Met by Problem Domain	
	Yes	No
Task solution has a high payoff	*	
Human expertise being lost	*	
Human expertise scarce	*	
Expertise needed in many location	*	
Expertise needed in hostile environment		*

Table 3

Characteristics That Make the Use of Expert Systems Appropriate Applied to TOs

Criteria	Criteria Met by Problem Domain	
	Yes	No
Task required symbol manipulation	*	
Task required heuristic solutions	*	
Task is not too easy	*	
Task has practical value	*	
Task is of manageable size	*	

Results of the Identification Phase

The first result of the identification phase was the selection of the domain expert and other participants. The domain expert selected by the knowledge engineer was Mr. O. J. Frazier, Chief of Technical Support Division, Logistics Directorate, Deputy for Aeronautical Equipment, Aeronautical Systems Division. Mr. Frazier was selected for his extensive experience in TOs. He has been a civil servant for ten years, with the last nine years in TO management at ASD. Prior to beginning civil service, Mr. Frazier retired from the Air Force as a Chief Master Sergeant after twenty-eight years as an avionics maintenance technician. While in the Air Force, he served in Strategic Air Command, Air Training Command, Tactical Air Command, United States Air Forces in Europe, and Pacific Air Forces. Along with his thirty-eight years experience as a manager and user of TOs, Mr. Frazier has an excellent reputation as an educator of young TO managers. He was willing to support the prototype development and set aside time to work with the knowledge engineer over a period of two weeks for approximately ninety minutes each day. In addition to the domain expert, the other participants were identified.

The additional TO experts selected to assist in evaluating the prototype were Mr. Riley Gust, Focal Point for TOs on the ASD Acquisition Logistics Staff, and Mr. Art Munguia, course director for the Air Force Institute of Technology TO Acquisition Management Course. To represent the intended users at ASD, Mrs. Marie Rotert, TO manager for the Advanced Tactical Air Reconnaissance System, and TSgt Michael Mires, TO manager for the AN/ALE-47 Chaff and Flare Dispenser, were selected to participate during the testing phase.

The second result of this phase was identification of the problem characteristics and scope. The problem domain was characterized by the goal of providing those using TOs with the most current and accurate information in a timely manner. To this end, the problem domain was divided into the areas of acquiring TOs, using TOs, and managing existing TOs. Focusing on the acquisition of TOs, a further decomposition of the problem resulted in narrowing the focus to the planning and execution of a TO acquisition program. A closer examination of the planning for TO acquisition highlighted the need for appropriate contractual documentation to implement The TO acquisition program. However, this contractual documentation, the Statement of Work (SOW), Contract Data Requirements List (CDRL) items, and the Technical Manual Contract Requirements (TMCR) 86-01, was dependent on information about the weapon system program. This information consisted of the program phase, maintenance concept, complexity of technology, requirement for source data (for aircraft installation), number of TOs being developed, classification of the TOs, and the type of program (new development, modification, or non-developmental). This information would be used to arrive at an acceptable solution for developing the contractual documentation for a TO acquisition program. The scope of the problem for

purposes of developing the prototype was determined to be the preparation of the contractual documentation. Developing the contractual documentation was considered to be of manageable size and practical value.

The third result of the identification phase centered on the sources of knowledge, time, and computing facilities needed to develop the prototype. The reference documents that proved to be of value in developing the prototype system were AFR 8-2 Air Force Technical Order System and TO 00-5-1 AF Technical Order System. Also, Mr. Frazier provided examples of TO programs for the ACES II Ejection Seat and the Precision Location Strike System. He continually drew on his past experience as a TO manager and an avionics maintenance technician for knowledge of TOs and TO acquisition. The knowledge engineer estimated that approximately fifteen hours would be required for knowledge acquisition and another three hours for Mr. Frazier to evaluate the prototype. The time required to develop and test the prototype system was estimated by the knowledge engineer to be six and two weeks, respectively. The knowledge engineer determined that the required computing resources would include a Zenith Z-248 or compatible system and an expert system development shell.

Lastly, the knowledge engineer and Mr. Frazier decided on the goals for the prototype system. The goals for the prototype system were established as formalizing an informal set of procedures and distributing scarce expertise to inexperienced TO managers in developing the contractual documentation for acquiring TOs. Consequently, the knowledge engineer and Mr. Frazier decided that the prototype system should be oriented towards small programs because of the number of inexperienced TO managers assigned to such programs. This small program orientation of

the prototype and the goals of the system influenced the form of the prototype's output. The desired output was determined to be the complete SOW verbiage, the entries for the DD Form 1423 of the CDRL, and the suggested tailoring of sections two and three (General and Specific Requirements, respectively) of the TMCR 86-01.

Results of the Conceptualization Phase

In the conceptualization phase, the knowledge engineer used unstructured interviewing to arrive at and expand on the key concepts and relation in developing the contractual documentation for TO acquisition. The key concepts and relations that were discussed in detail were the program phase covered by the contract, the maintenance concept, the complexity of the technology, the requirement for source data, number of TOs being developed, the classification of the TOs, and the type of program. Each of these concepts was examined to determine the relationship among the concepts and between the concepts and the contractual documentation, the output of the system.

Key Concepts. The program phase covered by the contract influences the decision on whether to acquire TOs and the type of TOs (formal or developmental). Five possibilities for the program phase emerged. The program phase possibilities are concept exploration, demonstration/validation, full scale development (FSD), FSD with production options, and production. In concept exploration and demonstration/validation, no TO specific inputs for the contract are required. In these phases, the logistics support analysis program is the means for gathering logistics data that will be used in developing the weapon system's TOs in the FSD and production phases. A contract that covers only FSD dictates that developmental manuals will be acquired.

Developmental manuals are TOs that are not in military specification format and have not been validated. Also, developmental manuals are not subject to in-process reviews. Rather, developmental manuals are reviewed and commented on by the Air Force one time prior to delivery. Consequently, the SOW requirements for in-process reviews and validation and the CDRL items for the Validation Plan and the Validation Completion Report can be deleted. Developmental manuals offer the advantages of cost savings (because of the relaxed requirements) and of a baseline for formal TOs should the weapon system go into production. In a program with a contract covering FSD with production options or production, formal TOs will be acquired to support the fielding of the weapon system. Regarding the contractual documentation, a program in production will require in-process reviews and validation tasks in the SOW and the corresponding data in the CDRL.

The maintenance concept determines the TOs that will be required to support the weapon system. The TOs must support the maintenance concept chosen by the user. The possible maintenance concepts are the following: throw-away (discard the item at failure); organic organizational (O) and intermediate (I) level maintenance; organic O and depot (D) level maintenance; organic O, I, and D level maintenance; and contractor logistics support (CLS). CLS does not require any specific TO inputs to the contractual documentation, because the contractor is responsible for the weapon system for the life of the weapon system. A weapon system with a throw-away maintenance concept requires TOs for the operation of the system. An organic O and I level maintenance concept requires TOs for the operation and maintenance of the weapon system. TOs for organizational level operation and maintenance and depot level overhaul support an organic O and D level maintenance concept. The maintenance

concept of organic maintenance at the O, I, and D level dictates TOs for the operation, maintenance, and overhaul of the weapon system. The TOs necessary to support the weapon system maintenance concept are reflected in the first paragraph of the TO section of the SOW.

The complexity of the technology in a weapon system determines the number of in-process reviews required to ensure that the TO development is progressing satisfactorily. A weapon system's technology can be characterized as simple, moderately complex, or highly complex. With simple technology, one in-process review conducted when the TOs are at the fifty percent completion point is sufficient. Two in-process reviews conducted at the thirty and seventy percent completion points are needed when the technology is moderately complex. A highly complex system necessitates three in-process reviews at the thirty, sixty, and ninety percent completion points in the TO development. The number of in-process reviews is stated in the first paragraph of the SOW.

Source data is required to support installation of a weapon system on an aircraft and aircraft interface. If a weapon system is going to be installed on an aircraft, the contractor is tasked to deliver source data in the SOW and CDRL, Technical Manual Research and Analysis Source Data. Otherwise, the requirement is omitted from the SOW and CDRL. Also, the requirement for source data indicates a need for Time Compliance Technical Orders (TCTOs). A TCTO is a TO that addresses the installation of a new weapon system on an aircraft or other equipment. The TMCR 86-01 elucidates the requirements for TCTOs.

The number of TOs being acquired influences the decision on which TO management tasks are to be imposed on the contractor. In the case of a weapon system program that is acquiring less than three TOs, the requirements for a Technical Manual Publication Plan, the Report of

Technical Manual Costs, and the Validation Plan are not levied in the SOW or CDRL. Dropping these requirements reduces the cost of acquiring the TOs without sacrificing adequate management visibility and control. The TO manager can adequately monitor a small number of TOs in development without these management tasks. On the other hand, a TO program that encompasses three or more TOs requires the management tasks (Technical Manual Publication Plan, Report of Technical Manual Costs, and Validation Plan) to ensure that the contractor is conducting the TO program properly. Therefore, these management tasks are required in the SOW and CDRL.

The type of weapon system program influences the tailoring of sections two and three of the TMCR 86-01. The types of weapon system programs are new development, non-developmental, and Class V modification (permanent change in the configuration of an existing weapon system). The type of program indicates the probable availability of commercial manuals and the need for TCTOs. A new development effort is unlikely to have commercial manuals for the system or equipment available; therefore, the requirements to address commercial manuals and TCTOs are tailored out of the TMCR. In contrast, a non-developmental weapon system indicates that the equipment is commercially available. As such, non-developmental systems usually have commercial manuals that support the equipment. A Class V modification to a weapon system requires a TCTO to explain the change in the system's configuration.

The possibility of classified TOs is addressed in the TMCR. Consequently, a weapon system that does not require that any classified information be included in the TOs can delete the paragraph on classified TOs from the TMCR. If the TOs are going to contain classified information, the paragraph would remain in the TMCR.

Relations Among the Key Concepts. After extracting the key concepts from Mr. Frazier, the knowledge engineer analyzed the relations among the key concepts. Table 4 shows the number of possibilities for each of the key concepts and the number of unique combinations of those possibilities. Applying the product rule for combinations, the result is that 1,800 unique combinations of the key concepts exist. Of the 1,800 unique combinations, those combinations that did not require any specific inputs to the contractual documentation were considered trivial. A combination was considered trivial if the program phase was concept exploration or demonstration/validation or if the maintenance concept was CLS. By reducing the number of possibilities for the program phase from five to three and the number of possible maintenance concepts from five to four, the number of unique non-trivial (meaning inputs to the contractual documentation are required) combinations was calculated, using the product rule again, to be 864.

Relations Between the Key Concepts and the Contractual Documentation. In analyzing the relations between the key concepts and the contractual documentation, the knowledge engineer examined how (if at all) a key concept affected the SOW, CDRL items, and the TMCR. The SOW was found to be affected by the program phase, maintenance concept, complexity of the technology, requirement for source data, and the number of TOs being acquired. Since the CDRL items are a reflection of the SOW, the CDRL items were found to be affected by all the same key concepts as the SOW. The tailoring of sections two and three of the TMCR were found to be affected by the type of program, requirement for source data, and the classification of the TOs.

After closer examination, the knowledge engineer learned that the first paragraph of the SOW was determined by the program phase,

Table 4

Number of Unique Combinations from the Key Concepts

Key Concept and Possibilities	Number of Possibilities
Program phase	5
Concept Exploration	
Demonstration/Validation	
FSD	
FSD with Production Options	
Production	
Maintenance Concept	
Throw Away	5
Organic O and I Level	
Organic O and D Level	
Organic O, I, and D Level	
CLS	
Complexity of the Technology	3
Simple	
Moderately Complex	
Highly Complex	
Source Data Required	2
Yes	
No	
Number of TOs Being Acquired	2
Less than Three	
Three or More	
Type of Program	3
New Development	
Class V Modification	
Non-Developmental	
TOs Contain Classified Information	2
Yes	
No	
Number of Unique Combinations from the Key Concepts	1800

maintenance concept, and the complexity of the technology. In addition, the only distinction made between program phases was whether the phase addressed any type of production provisions. Consequently, no distinction was made between the FSD with production options and production program phases. This narrowed the relevant program phases to

production-related (which encompasses FSD with production options and production) and FSD. Further, the first paragraph of the SOW for a program in FSD was affected by the maintenance concept only, not the complexity of the technology. The complexity of the technology is not a factor for programs in FSD because in-process reviews are not required in strictly FSD programs. In the case of production-related program phase, the first paragraph of the SOW was found to be affected by both the maintenance concept and the complexity of the technology. The result was that sixteen unique combinations, summarized in Table 5, can cover the first paragraph of the SOW. The remainder of the SOW was determined by the requirement for source data and the number of TOs being acquired. Therefore, the remainder of the SOW can be covered by four unique combinations as shown in Table 6. Since the first paragraph of the SOW does not reference any CDRL items, all the CDRL items are referenced by the remaining paragraphs in the SOW. Accordingly, the CDRL items required by the SOW can be covered by the same combinations as those for the remainder of the SOW (see Table 6).

The tailoring of sections two and three of the TMCR is dependent on the type of program, requirement for source data, and classification of the TOs. The only peculiarity in ascertaining the tailoring for the TMCR is that a Class V modification program always requires source data. Consequently, the tailoring of the TMCR can be covered by the ten unique combinations summarized in Table 7.

The relations between the key concepts and the contractual documentation can be described by the unique combinations shown in Tables 5, 6, and 7. In turn, the combinations from the first paragraph of the SOW, remainder of the SOW and CDRL items, and tailoring of the TMCR can be combined to yield the number of unique combinations of contractual

Table 5

Unique Combinations for the First Paragraph of the SOW

Program Phase	Maintenance Concept	Complexity
FSD	Throw-Away	N/A
FSD	Organic O and I	N/A
FSD	Organic O and D	N/A
FSD	Organic O, I, and D	N/A
Production-Related	Throw-Away	Simple
Production-Related	Throw-Away	Moderately Complex
Production-Related	Throw-Away	Highly Complex
Production-Related	Organic O and I	Simple
Production-Related	Organic O and I	Moderately Complex
Production-Related	Organic O and I	Highly Complex
Production-Related	Organic O and D	Simple
Production-Related	Organic O and D	Moderately Complex
Production-Related	Organic O and D	Highly Complex
Production-Related	Organic O, I, and D	Simple
Production-Related	Organic O, I, and D	Moderately Complex
Production-Related	Organic O, I, and D	Highly Complex

Table 6

Unique Combinations for the Remainder of the SOW

Source Data Required	Number of TOs Being Acquired
Yes	Less than Three
Yes	Three or More
No	Less than Three
No	Three or More

Table 7

Unique Combinations for the Tailoring of the TMCR

Type of Program	Source Data Required	Classified TOs
New Development	Yes	Yes
New Development	Yes	No
New Development	No	Yes
New Development	No	No
Non-Developmental	Yes	Yes
Non-Developmental	Yes	No
Non-Developmental	No	Yes
Non-Developmental	No	No
Class V Modification	Yes	Yes
Class V Modification	Yes	No

documentation possible. These unique combinations of the contractual documentation address all the non-trivial combinations from the key concepts (see Table 4). Applying the product rule, there are 640 unique combinations of the contractual documentation (sixteen unique combinations for the first paragraph of the SOW, four unique combinations for the remainder of the SOW and CDRL items, and ten unique combinations for tailoring the TMCR) required to address the 864 non-trivial unique combinations from the key concepts. The unique combinations of the contractual documentation is shown in Table 8. Having arrived at a conceptualization of the key concepts and the relations among the key concepts and between the key concepts and the contractual documentation, the knowledge engineer had to choose a knowledge representation scheme and a development shell.

Table 8

Number of Unique Combinations from the Contractual Documentation

Contractual Documentation	Number of Possibilities
First Paragraph of the SOW	16
Remainder of the SOW and CDRL Items	4
Tailoring of the TMCR	10
Number of Unique Combinations from the Contractual Documentation	640

Results of the Formalization Phase

The formalization phase yielded two results. First, rules were determined to be the most appropriate knowledge representation scheme. Second, VP-EXPERT was selected as the shell to use in developing the prototype system.

In choosing a representation scheme, the knowledge engineer rejected semantic networks and frames. The links among the key concepts were not

significant enough to take advantage of the semantic network's ability to represent links between concepts. In addition, the need to show inheritance of properties from a higher level concept to a lower level concept or to classify domain knowledge was not evident during the conceptualization and identification phases. The conceptualization and identification phases did not determine that the form and content of the data was crucial to arriving at the contractual documentation for TOs. Therefore, the problem domain of TOs is not similar to the domains which use frames as a representation scheme. Frames did offer the advantage of being able to represent the contractual documentation as nodes. Each node could then be described by slots for the key concepts that determine the required contractual documentation. However, both frames and semantic networks did not match the key concepts and minimize the representational mismatch as well as rules.

The determining factor in the selection of rules as the representation scheme was Mr. Frazier's continual use of the phrase "a lot of this [determining the appropriate contractual documentation] is just experience" during the interviews. This phrase alerted the knowledge engineer to the fact that a large amount of Mr. Frazier's knowledge had been the result of empirical associations developed during his thirty-eight years in working with TOs. Such observed relationships are well suited to representation through rules. Also, each piece of the contractual documentation could be inferred from the facts about the relevant key concepts. This inference strategy supported backward chaining as an efficient (as compared to forward chaining) and effective inference method.

After rejecting programming languages as a development tool, the knowledge engineer selected VP-EXPERT as the shell to be used in

implementing the prototype system. The selection of VP-EXPERT was made based on several criteria. First, VP-EXPERT matches the characteristics of the problem domain and supports rules as a knowledge representation scheme. In addition, VP-EXPERT performs forward and backward chaining through the knowledge base. The training time to learn how to use the shell was approximately three days and the shell has an extensive on-line help system. Lesser factors in selecting VP-EXPERT were its ability to accommodate uncertainty factors, the ease of user interface for the prototype system (VP-EXPERT supports text dialogue and pop-up menus), and the low cost of the shell (approximately $100). Finally, VP-EXPERT supported the only "must have" feature for selection - compatibility with the Zenith Z-248.

Results of the Implementation Phase

During the implementation phase, the knowledge engineer wrote the program code for the prototype system. Before actually beginning to write code, the knowledge engineer decided on the variables and the contents of the variables. Also, the inference and control strategy was determined prior to generating any code. The form of the program was dictated by the shell and its rules for format and syntax rules.

The variables were created to match the key concepts from the conceptualization and identification phases. Variables were created to correspond with the program name, program phase, maintenance concept, complexity of the technology, requirement for source data, number of TOs being acquired, type of program, and the classification of the TOs. The contents of each variable is given by the user. The user determines the contents of each variable, except for the program name, by selecting from a system provided menu of choices. The program name is entered

arbitrarily by the user and is only to make the prototype appear friendlier. The menu for each variable reflects the possibilities for that variable (see Table 4). The variables are prompted in an order that is natural and progressive, which makes the user more confident in the prototype's competency. In addition to the variables for the key concepts, intermediate variables were created. For example, a single intermediate variable may reflect that the program phase is production and the maintenance concept is throw-away. The contents of the intermediate variable is then used in conjunction with the contents of the variable for the complexity of the technology to arrive at the appropriate first paragraph of the SOW.

The selection of rules as the knowledge representation scheme limited the inference and control strategies to forward and backward chaining. VP-EXPERT supports both inference methods. However, since each piece of the contractual documentation could be inferred from the relevant key concepts, the knowledge engineer chose backward chaining as the inference and control strategy.

The actual program code was the result of approximately six weeks of coding, testing, modifying, and re-testing. The first attempt at generating the program code was a failed effort. This first attempt centered on the use of the shell's ability to generate rules from an induction table. The knowledge engineer built an induction table, using a spreadsheet program, to reflect all the unique combinations for the key concepts and the contractual documentation. While this effort did produce accurate and usable code, the code was to unwieldy and inefficient. At one point, the rule base exceeded the working memory limits and the rule base had to be divided into three rule bases. The total number of rules in these three rule bases was approximately 270.

The knowledge engineer was handicapped by previous experience in coding for conventional programs. Negative learning transfer occurred as the knowledge engineer was actually attempting to code in a brute force conventional manner, unconsciously applying conventional coding techniques. After two weeks of coding, the knowledge engineer discovered a different perspective in the coding and began to understand how to code for a backward chaining expert system. The number of rules was reduced from 270 to 52. The efficiency of the prototype increased as well since now the minimum number of rules were executed to determine the contractual documentation required for the TO program. Also, the code was written to give the user an explanation as to why and how the contents of the variable are important to the contractual documentation. The knowledge engineer spent four weeks writing the code, testing the code, modifying the code as necessary, and re-testing the code until the code was believed to be accurate and adequate. The complete listing of the code is in Appendix A.

<u>Results of the Testing Phase</u>

The testing phase was a two part evaluation. The first part was to assess the accuracy of the prototype system's knowledge and performance. This part of testing was conducted by the knowledge engineer and Mr. Frazier. The second part of the testing phase was aimed at validating the prototype performance by Mr. Gust and Mr. Munguia and assessing the utility of the prototype by Mrs. Rotert and TSgt Mires.

The first step in Mr. Frazier's evaluation of the prototype involved going through the knowledge base and checking the prototype's reasoning processes. This initial check of the knowledge base did not reveal any deficiencies in the prototype. Also, Mr. Frazier compared the

prototype's inference and control strategies to his own strategies. Again, no discrepancies were found. The next step was to use the prototype to determine the contractual documentation for two weapon system programs. The weapon system programs were programs that Mr. Frazier had prepared the contractual documentation for the TOs. The programs were the LPU/9 Life Preserver and the Mark XV Identification Friend or Foe (IFF). The output from the prototype (the output for the Mark XV IFF is in Appendix B) was compared to the contractual documentation that Mr. Frazier had prepared for the two weapon systems. The output was not accurate in omitting the Validation Plan if the number of TOs being acquired is less than three and in tailoring paragraph 17.2 of the TMCR to match the maintenance concept. The knowledge engineer corrected these two problems and the prototype performance was considered accurate and the output adequate. Table 9 shows Mr. Frazier's answers to the specific questions posed by the knowledge engineer at the end of this stage of testing. Mr. Frazier's concluding comments on the prototype were of his views on the limitations and utility of the prototype. He sees the prototype, in its present state, as limited to small programs and as useful to inexperienced TO managers faced with the task of preparing a SOW, CDRL items, and TMCR tailoring for a TO acquisition effort.

The second part of the testing began with an evaluation of the prototype by Mr. Gust. He reviewed the output generated by the prototype for the LPU/9 and the Mark XV IFF. Mr. Gust stated that he "generally" agreed with the output of the prototype for those two weapon systems. He then created a scenario for a fictitious weapon system program and reviewed the prototype's output for that fictitious weapon system program. While he "generally" agreed with the output of the

Table 9

Domain Expert's Answers on Prototype Performance

Question	Answer
Was the system's output (advice and decisions) appropriate?	Yes
Is the knowledge base correct, consistent, and complete?	Yes
Do the control strategies consider items in the natural order?	Yes
Are the explanations of the "how and "why" of actions and conclusions adequate?	Yes

prototype, his strongest objection to the prototype was its use of Class V modification as a choice the type of program. Mr. Gust asserted that "Class V modification" is a poor choice of words that will confuse new TO managers over who is responsible for writing TCTOs. Also, Mr. Gust contends that ASD does not usually get involved in Class V modification, which are usually the Air Logistic Centers' responsibility. In fact, he expressed reservations over the need for the variable concerning the type of program at all. The other aspect of the prototype Mr. Gust disagreed with was the deletion of the Technical Manual Publication Plan and the Report of Technical Manual Costs when acquiring less than three TOs. His disagreement centered on his view that deleting the Technical Manual Publication Plan should not be based solely on the number of TOs being acquired. Mr. Gust considers the number of TO pages and the complexity of the TOs as factors in deciding to delete the Technical Manual Publication Plan. Furthermore, he does not concur with deleting the Report of Technical Manual Costs under any circumstances. Mr. Gust does not think the TO manager has the information needed to ascertain the contractor's expenditures in developing a TO. Despite his objections,

Mr. Gust said that "the concept [expert system for TO acquisition] is something ASD could use and it needs to be expanded."

The other TO expert to evaluate the prototype was Mr. Munguia. Like Mr. Gust, Mr. Munguia "generally" agreed with the prototype's output for the LPU/9 and the Mark XV IFF. He created a scenario for a fictitious weapon system and reviewed the prototype's output for that fictitious weapon system. Mr. Munguia did not think the prototype provided enough information to the TO manager when preparing to transition a weapon system from demonstration/validation to FSD. He suggested that the prototype recommend actions to the TO manager in order to prepare for FSD, the first program phase that has specific TO tasks. In addition, Mr. Munguia objected to deleting the requirements for TCTOs from the TMCR for any weapon system that is in a production-related phase. He sees that as not providing the Air Force a method to handle configuration changes that may arise during production. His final point of contention with the prototype was the reference to data in the SOW verbiage. Mr. Munguia stated that this is an occasional problem with contracting officers who oppose references to data in the SOW text. Concerning the prototype and its output, he said that the "information is very informative and helpful to the new TO manager." Specific questions about the prototype's performance were asked of both additional experts. The questions and the additional experts' answers to those questions are shown in Table 10.

The final part of the testing was conducted by Mrs. Rotert and TSgt Mires. They represented the typical end-user of the prototype and evaluated the utility of the prototype. Mrs. Rotert was first to evaluate the prototype. Following a brief explanation of the prototype, the knowledge engineer worked through an example using the prototype as

Mrs. Rotert observed. Subsequently, she used the prototype and keyed the appropriate responses for her weapon system program, the Advanced Tactical Air Reconnaissance System. Mrs. Rotert reviewed the prototype system's output and answered specific questions about the utility of the prototype. TSgt Mires's evaluation of the prototype followed the same sequence of events as Mrs. Rotert's evaluation. The only difference being that TSgt Mires used his weapon system program, the AN/ALE-47 Chaff and Flare Dispenser. Specific questions about the prototype's utility wee asked of both TSgt Mires and Mrs. Rotert. The questions and their answers to those questions are shown in Table 11.

Summary

This chapter analyzed and recorded the results of the research conducted in developing the prototype system. The results of each step in the development methodology (selection of the problem domain, identification, conceptualization, formalization, implementation, and testing) were discussed in this chapter. The next chapter addresses the conclusion and recommendations of the research.

Table 10

Additional Experts' Answers on Prototype Performance

Question	Answers: Mr. Gust	Answers: Mr. Munguia
Was the system's output (advice and decisions) appropriate?	For a small program, the output is adequate. However, the output may have modified by the TO manager.	Yes, generally speaking it provides the basic requirements and guides the inexperienced TO manager.
Do the control strategies consider items in the natural order?	Normal sequence.	Did not see any problems.
Are the explanations of the "how" and "why" of actions and conclusions adequate?	For the explanation on the complexity, expand on the role of TO content in in-process reviews. For the explanation on source data, the requirement for TCTOs is not always firm.	Yes.
What changes to the system would you recommend?	Expand the system to address modifications less than Class V and tailoring of sections four and five of the TMCR. Delete the variable for the type of program. Add coverage of digital TOs in the output.	Enhance the prototype by adding tailoring for sections four and five of the TMCR and for MIL-STD-1790A. Add a statement to the discussion of the demonstration/validation phase to alert the TO manager to begin preparing for FSD. Add a statement clarifying the fact that the output is inclusion in the Request for Proposal for the upcoming program phase.

Table 11

End-Users Answers on Prototype Utility

Question	Answers: Mrs. Rotert	Answers: TSgt Mires
Is the system output helpful in a meaningful and significant way?	Helpful to someone just starting out, because it does not just say fill in these squares. It says here's why these squares are important.	Saves time in developing the contractual documentation. Good training tool for new TO managers.
Is the output organized well and presented at the right level of detail?	Fine, some of the information on the CDRL items duplicates the Data Item Description. The output progresses logically.	Well organized. The SOW reiterated some of the information found in the Data Item Descriptions.
Does the system's response time seem excessive?	No.	No, it is quick.
Is the system interface friendly enough?	Very easy, the instructions are clear and complete. Nothing is complicated.	Straightforward, the instructions are clear.
What changes to the system would you recommend?	Extend the tailoring of the TMCR to section four. Expand the explanations for training purposes.	None.

V. Conclusions and Recommendations

Overview

This chapter covers the conclusions and recommendations of the research. Each section of this chapter answers one of the research questions posed in chapter one. The sections address the following: the applicability of expert system technology to TO acquisition tasks; the required resources, participants, goals, and problem characteristics for the prototype; the key concept and relations in the selected domain of TO acquisition; the appropriate knowledge representation scheme and development tool; the required data structures and control strategies; and the competency and utility of the prototype. The final section of the chapter discusses recommendations for the prototype system.

Suitable TO Acquisition Tasks for Expert System Application

The first research question posed in chapter one was "What TO acquisition tasks are suitable for expert system application?" The conclusion is that both planning and executing a TO acquisition program are suitable tasks for expert system technology. However, the knowledge engineer and the domain expert considered the task of preparing the contractual documentation to be of the most practical value while still being manageable in size. Therefore, the prototype was developed to provide the contractual documentation necessary to execute a TO acquisition program.

Resources, Participants, Goals, and Problem Characteristics

The second research question posed in chapter one was "What are the required resources, necessary participants, appropriate goals, and

problem characteristics for a prototype system for TO acquisition?" The answer to this question is composed of several parts. The required resources for developing a prototype system for TO acquisition are pertinent literature on TO acquisition, a willing TO expert, approximately ten weeks of the knowledge engineer's time (two weeks for knowledge acquisition, six weeks for coding the program, and two weeks for evaluating the prototype), and access to computing facilities and expert system development tools. The participants required to develop a prototype system are a TO expert(s) to build the prototype, additional TO experts to validate the prototype's performance, and typical end-users to assess the prototype's utility. The appropriate goals for a prototype system for TO acquisition are formalizing an informal set of procedures and distributing scarce expertise to inexperienced TO managers in developing the contractual documentation for acquiring TOs. The problem domain was characterized by the use of the weapon system program's attributes to provide TOs with the most current and accurate information in a timely manner.

Key Concepts and Relations

"What are the key concepts and relations used in TO acquisition?" was the third research question asked in chapter one. The conclusion is that the key concepts in TO acquisition are the program phase, maintenance concept, complexity of the technology, requirement for source data, number of TOs being acquired, type of program, and the classification of the TOs. These key concepts relate to the contractual documentation for TO acquisition. The program phase, maintenance concept, and complexity of the technology influence the first paragraph of the SOW. The remainder of the SOW and the CDRL items are determined by the requirement for source data and the number of TOs being acquired.

The type of program, classification of the TOs, and the requirement for source data influence the tailoring of sections two and three of the TMCR 86-01.

Appropriate Knowledge Representation Scheme and Development Tool

The fourth research question asked in chapter one was "What is the appropriate knowledge representation scheme and tool for developing the prototype system?" The answer to this question is that a rule based scheme is the most appropriate knowledge representation scheme for TO acquisition. However, the use of frames is not without merit as an alternative representation scheme. The most appropriate tool for developing the prototype is VP-EXPERT. This shell meets all the criteria put forth for selection and offers the ability to expand the prototype.

Data Structures and Control Strategies

"What data structures and control strategies are required in the prototype system?" was the fifth research question posed in chapter one. The conclusion is that the data structures should represent the key concepts. Also, each data structure should contain a menu of choices that corresponds to the possibilities for each key concept. The prototype's control strategy is backward chaining through the rule base. This control strategy produced the most efficient and effective method for drawing conclusions about the contractual documentation.

Competency and Utility of the Prototype

The sixth and last research question asked in chapter one was "How competent and useful is the prototype?" The conclusion is that the prototype is competent in determining the contractual documentation for TO acquisition. While the domain expert, Mr. Frazier, agreed with the output of the prototype, the additional experts, Mr. Gust and Mr.

Munguia, agreed with the output in only a general sense. Each of the additional experts would change the output of the prototype to reflect his problem solving approach. The less than total validation of the system by the additional experts is attributed to the different problem solving approaches among the different TO experts. From the standpoint of the prototype's utility, the prototype is judged as useful in helping inexperienced TO managers in preparing the contractual documentation and in training. The representative users, Mrs. Rotert and TSgt Mires, view the prototype system as helpful and easy to use.

Recommendations

The recommendations address the suggested use of the prototype and further research. The prototype should be sent to ASD for use on a day-to-day basis by TO managers in the system program offices. In fact, Mr. Frazier, Mrs. Rotert, and TSgt Mires expressed interest in obtaining a copy of the prototype. Also, Mr. Gust is considering a demonstration of the prototype at the next ASD TO workshop. Such day-to-day usage would serve to test the system further and discover its strengths and weaknesses.

On a larger scale, the application of expert system technology to other areas of weapon system acquisition is warranted. Most of the functional areas of weapon system acquisition are prime candidates for expert system applications. Prototype expert systems for those areas should be developed.

Research into expanding the prototype system for TO acquisition could include adding other TO experts' knowledge to the knowledge base. By adding other experts' knowledge, the prototype could include other problem solving approaches and expertise about subdomains (such as TOs for non-developmental equipment). Also, research into tailoring sections

four and five of the TMCR would enlarge the ability of the system to provide more comprehensive contractual documentation. The last recommendation is that the explanations for the actions and conclusions be expanded to serve as a training aid for inexperienced TO managers.

Summary

This chapter covered the conclusions and recommendations of the research. The first six sections of this chapter answered the research questions posed in chapter one. The sections discussed the conclusions for the following: the applicability of expert system technology to TO acquisition tasks; the required resources, participants, goals, and problem characteristics for the prototype; the key concept and relations in the selected domain of TO acquisition; the appropriate knowledge representation scheme and development tool; the required data structures and control strategies; and the competency and utility of the prototype. The last section of the chapter addressed recommendations for the use of the prototype system and suggested areas for further research.

Appendix A: Program Code for the Prototype Expert System

```
!FILENAME:  TECHORDR.KBS
!
!PROGRAMMER:  Capt Jim Harvell, AFIT/LSA, AV785-5435
!
!DATE:  6 Jul 88
!
!PURPOSE:  The purpose of this prototype expert system is to
!           demonstrate the feasibility of using expert system
!           technology in technical order (TO) acquisition.  This
!           system is aimed at the TO managers working on small
!           programs.  The system does not address the acquisition
!           of TOs for very large, stand-alone programs such as
!           the F-16, B-1B, or Ground Launched Cruise Missile.
!           The TO managers on small programs are often
!           inexperienced and/or tasked with managing more
!           logistics elements than just TOs.
!
!           The prototype system prompts the user for information
!           concerning the program then the system suggests
!           Statement of Work paragraphs, Contract Data
!           Requirements List items, and tailoring of Sections 2
!           and 3 of Technical Manual Contract Requirements 86-01.


EXECUTE;
RUNTIME;
ENDOFF;

ACTIONS
        CLROFF
        DISPLAY
"This prototype system is designed to assist ASD tech order
managers of small programs in preparing the contractual
documentation needed to execute a tech order acquisition.  A
small program is defines as a program that is not a large,
stand-alone weapon system.  For example, the a system going onto
an aircraft is considered a small program, but a new aircraft is
a large system.  You will be asked a series of questions
concerning the program.  The system will then use this
information to suggest Statement of Work paragraphs, Contract
Data Requirements List items, and tailoring of Sections 2 and 3
of Technical Manual Contract Requirements 86-01."
        DISPLAY
" "
        DISPLAY
"USE THE ARROW KEYS TO HIGHLIGHT YOUR ANSWER AND PRESS ENTER."
        DISPLAY
" "
        DISPLAY
"THESE SUGGESTIONS WILL BE SENT TO YOUR PRINTER SO MAKE SURE THE
```

```
PRINTER IS ON AND AT THE TOP OF THE PAGE."
        DISPLAY
 " "
        DISPLAY
 "PRESS ANY KEY TO START~"
        CLS
        FIND Program_Name
        FIND First_SOW_Paragraph
        FIND Rest_of_SOW_and_CDRLs
        FIND Tmcr_Sections
        CLS
        DISPLAY
 " "
        DISPLAY
"THIS ENDS THE CONSULTATION.
PRESS ANY KEY TO RETURN TO THE MAIN MENU.~";

RULE 10
IF      Program_Phase=FSD_and_Prod_Options OR
        Program_Phase=Production
THEN    Contract=Production_Type
BECAUSE
"The period of performance on the contract influences the
decision whether to buy formal TOs (for any production type
contracts) or developmental manuals (for FSD contracts) or not
to buy any TOs (concept exploration and dem/val contracts).";

RULE 20
IF      Contract=Production_Type AND
        Maintenance_Concept=Throw_Away
THEN    Contract_and_Maintenance=Production_and_Throw_Away
BECAUSE
"The maintenance concept is the key in deciding what TOs will be
required to support the system.  The TOs must support the
maintenance concept chosen by the user.";

RULE 30
IF      Contract=Production_Type AND
        Maintenance_Concept=Organic_O_and_I
THEN    Contract_and_Maintenance=Production_and_Organic_O_and_I;

RULE 40
IF      Contract=Production_Type AND
        Maintenance_Concept=Organic_O_and_D
THEN    Contract_and_Maintenance=Production_and_Organic_O_and_D;

RULE 50
IF      Contract=Production_Type AND
        Maintenance_Concept=Organic_O_I_and_D
THEN    Contract_and_Maintenance=Production_and_Organic_O_I_and_D;

RULE 60
IF      Contract_and_Maintenance=Production_and_Throw_Away AND
        Complexity=Simple
THEN    First_SOW_Paragraph=SOW11
```

```
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow11
BECAUSE
"The complexity of the system determines the number of in-process
reviews required to ensure the TOS are on track regarding
schedule and content.";

RULE 70
IF      Contract_and_Maintenance=Production_and_Throw_Away AND
        Complexity=Moderately_Complex
THEN    First_SOW_Paragraph=SOW12
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow12;
RULE 80
IF      Contract_and_Maintenance=Production_and_Throw_Away AND
        Complexity=Highly_Complex
THEN    First_SOW_Paragraph=SOW13
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow13;

RULE 90
IF      Contract_and_Maintenance=Production_and_Organic_O_and_I AND
        Complexity=Simple
THEN    First_SOW_Paragraph=SOW14
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow14
BECAUSE
"The complexity of the system determines the number of in-process
reviews required to ensure the TOS are on track regarding
schedule and content.";

RULE 100
IF      Contract_and_Maintenance=Production_and_Organic_O_and_I AND
        Complexity=Moderately_Complex
THEN    First_SOW_Paragraph=SOW15
```

```
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow15;

RULE 110
IF      Contract_and_Maintenance=Production_and_Organic_O_and_I AND
        Complexity=Highly_Complex
THEN    First_SOW_Paragraph=SOW16
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow16;

RULE 120
IF      Contract_and_Maintenance=Production_and_Organic_O_and_D AND
        Complexity=Simple
THEN    First_SOW_Paragraph=SOW17
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow17
BECAUSE
"The complexity of the system determines the number of in-process
reviews required to ensure the TOS are on track regarding
schedule and content.";

RULE 130
IF      Contract_and_Maintenance=Production_and_Organic_O_and_D AND
        Complexity=Moderately_Complex
THEN    First_SOW_Paragraph=SOW18
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow18;

RULE 140
IF      Contract_and_Maintenance=Production_and_Organic_O_and_D AND
        Complexity=Highly_Complex
THEN    First_SOW_Paragraph=SOW19
        DISPLAY
"
```

```
PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow19;

RULE 150
IF      Contract_and_Maintenance=Production_and_Organic_O_I_and_D AND
        Complexity=Simple
THEN    First_SOW_Paragraph=SOW110
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ..."
        BCALL sow110
BECAUSE
"The complexity of the system determines the number of in-process
reviews required to ensure the TOS are on track regarding
schedule and content.";

RULE 160
IF      Contract_and_Maintenance=Production_and_Organic_O_I_and_D AND
        Complexity=Moderately_Complex
THEN    First_SOW_Paragraph=SOW111
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow111;

RULE 170
IF      Contract_and_Maintenance=Production_and_Organic_O_I_and_D AND
        Complexity=Highly_Complex
THEN    First_SOW_Paragraph=SOW112
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL sow112;

RULE 180

IF      Program_Phase=FSD
THEN    Contract=FSD_Type;

RULE 190
IF      Contract=FSD_Type AND
        Maintenance_Concept=Throw_Away
THEN    First_SOW_Paragraph=FSDSOW1
        DISPLAY
"
```

```
PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL fsdsow1
BECAUSE
"The maintenance concept is the key in deciding what TOs will be
required to support the system.  The TOs must support the
maintenance concept chosen by the user.";

RULE 200
IF      Contract=FSD_Type AND
        Maintenance_Concept=Organic_O_and_I
THEN    First_SOW_Paragraph=FSDSOW2
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL fsdsow2;

RULE 210
IF      Contract=FSD_Type AND
        Maintenance_Concept=Organic_O_and_D
THEN    First_SOW_Paragraph=FSDSOW3
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL fsdsow3;

RULE 220
IF      Contract=FSD_Type AND
        Maintenance_Concept=Organic_O_I_and_D
THEN    First_SOW_Paragraph=FSDSOW4
        DISPLAY
"

PRINTING THE FIRST PARAGRAPH OF THE STATEMENT OF WORK INPUT ...

"
        BCALL fsdsow4;

RULE 230
IF      Contract=Production_Type AND
        Source_Data=Yes
THEN    Contract_and_Source_Data=Production_and_Yes
BECAUSE
"If the system/equipment is going to be installed on an aircraft
then data on the installation and aircraft interface will have
to be sent to the agency responsible for writing the aircraft
TOs.  Anytime source data is required, a Time Compliance Tech
Order (TCTO) is usually required as well.";
```

```
RULE 240
IF      Contract_and_Source_Data=Production_and_Yes AND
        Number_of_TOs=Less_than_Three
THEN    Rest_of_SOW_and_CDRLs=PSCYBK
        DISPLAY
"

PRINTING THE REMAINING PARAGRAPHS OF THE STATEMENT OF WORK INPUT
AND THE CONTRACT DATA REQUIREMENTS LIST ITEMS ...

"
        BCALL sowybk
        EJECT
        BCALL cdrlybk
        EJECT
BECAUSE
"If the number of TOs being acquired is less than three then the
TOMA can write the Technical Manual Publication Plan instead of
the contractor (resulting in cost savings).  With a limited
number of TOs, the TOMA can track the cost of each TO from
the contractor instead of receiving the Report of Technical
Manual Costs data item.";

RULE 250
IF      Contract_and_Source_Data=Production_and_Yes AND
        Number_of_TOs=Three_or_More
THEN    Rest_of_SOW_and_CDRLs=PSCYBKS
        DISPLAY
"

PRINTING THE REMAINING PARAGRAPHS OF THE STATEMENT OF WORK INPUT
AND THE CONTRACT DATA REQUIREMENTS LIST ITEMS ...

"
        BCALL sowybks
        EJECT
        BCALL cdrlybks
        EJECT;

RULE 260

IF      Contract=Production_Type AND
        Source_Data=No
THEN    Contract_and_Source_Data=Production_and_No;

RULE 270
IF      Contract_and_Source_Data=Production_and_No AND
        Number_of_TOs=Less_than_Three
THEN    Rest_of_SOW_and_CDRLs=PSCNBK
        DISPLAY
"

PRINTING THE REMAINING PARAGRAPHS OF THE STATEMENT OF WORK INPUT
AND THE CONTRACT DATA REQUIREMENTS LIST ITEMS ...

"
```

```
        BCALL sownbk
        EJECT
        BCALL cdrlnbk
        EJECT
BECAUSE
"If the number of TOs being acquired is less than three then the
TOMA can write the Technical Manual Publication Plan instead of
the contractor (resulting in cost savings).  With a limited
number of TOs, the TOMA can track the cost of each TO from
the contractor instead of receiving the Report of Technical
Manual Costs data item.";

RULE 280
IF      Contract_and_Source_Data=Production_and_No AND
        Number_of_TOs=Three_or_More
THEN    Rest_of_SOW_and_CDRLs=PSCNBKS
        DISPLAY
"

PRINTING THE REMAINING PARAGRAPHS OF THE STATEMENT OF WORK INPUT
AND THE CONTRACT DATA REQUIREMENTS LIST ITEMS ...

"
        BCALL sownbks
        EJECT
        BCALL cdrlnbks
        EJECT;

RULE 290
IF      Contract=FSD_Type AND
        Source_Data=Yes
THEN    Contract_and_Source_Data=FSD_and_Yes
BECAUSE
"If the system/equipment is going to be installed on an aircraft
then data on the installation and aircraft interface will have
to be sent to the agency responsible for writing the aircraft
TOs.  Anytime source data is required, a Time Compliance Tech
Order (TCTO) is usually required as well.";

RULE 300
IF      Contract_and_Source_Data=FSD_and_Yes AND
        Number_of_TOs=Less_than_Three
THEN    Rest_of_SOW_and_CDRLs=FSCYBK
        DISPLAY
"

PRINTING THE REMAINING PARAGRAPHS OF THE STATEMENT OF WORK INPUT
AND THE CONTRACT DATA REQUIREMENTS LIST ITEMS ...

"
        BCALL fsybk
        EJECT
        BCALL fcybk
        EJECT
BECAUSE
```

```
"If the number of TOs being acquired is less than three then the
TOMA can write the Technical Manual Publication Plan instead of
the contractor (resulting in cost savings). With a limited
number of TOs, the TOMA can track the cost of each TO from
the contractor instead of receiving the Report of Technical
Manual Costs data item.";

RULE 310
IF      Contract_and_Source_Data=FSD_and_Yes AND
        Number_of_TOs=Three_or_More
THEN    Rest_of_SOW_and_CDRLs=FSCYBKS
        DISPLAY
"

PRINTING THE REMAINING PARAGRAPHS OF THE STATEMENT OF WORK INPUT
AND THE CONTRACT DATA REQUIREMENTS LIST ITEMS ...

"
        BCALL fsybks
        EJECT
        BCALL fcybks
        EJECT;

RULE 320
IF      Contract=FSD_Type AND
        Source_Data=No
THEN    Contract_and_Source_Data=FSD_and_No;

RULE 330
IF      Contract_and_Source_Data=FSD_and_No AND
        Number_of_TOs=Less_than_Three
THEN    Rest_of_SOW_and_CDRLs=FSCNBK
        DISPLAY
"

PRINTING THE REMAINING PARAGRAPHS OF THE STATEMENT OF WORK INPUT
AND THE CONTRACT DATA REQUIREMENTS LIST ITEMS ...

"
        BCALL fsnbk
        EJECT
        BCALL fcnbk
        EJECT
BECAUSE
"If the number of TOs being acquired is less than three then the
TOMA can write the Technical Manual Publication Plan instead of
the contractor (resulting in cost savings). With a limited
number of TOs, the TOMA can track the cost of each TO from
the contractor instead of receiving the Report of Technical
Manual Costs data item.";

RULE 340
IF      Contract_and_Source_Data=FSD_and_No AND
        Number_of_TOs=Three_or_More
THEN    Rest_of_SOW_and_CDRLs=FSCNBKS
```

```
        DISPLAY
"

PRINTING THE REMAINING PARAGRAPHS OF THE STATEMENT OF WORK INPUT
AND THE CONTRACT DATA REQUIREMENTS LIST ITEMS ...

"
        BCALL fsnbks
        EJECT
        BCALL fcnbks
        EJECT;

RULE 350
IF      Type_of_Program = New_Development AND
        Source_Data=Yes
THEN    Program_Type_and_Source_Data=New_Development_and_Yes
BECAUSE
"The type of program indicates the probable availability of
commercial manuals and need for Time Compliance Technical Orders.
This information is used in tailoring the TMCR 86-01.";

RULE 360
IF      Program_Type_and_Source_Data=New_Development_and_Yes AND
        Classified = Yes
THEN    Tmcr_Sections = TMCR1
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 86-01 ...

"
        BCALL tmcr1
        EJECT
BECAUSE
"If the TOs will not have to be classified then the paragraph on
classified TOs can be tailored out of the TMCR 86-01.";

RULE 370
IF      Program_Type_and_Source_Data=New_Development_and_Yes AND
        Classified = No
THEN    Tmcr_Sections = TMCR3
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 86-01 ...

"
        BCALL tmcr3
        EJECT;

RULE 380
IF      Type_of_Program = New_Development AND
        Source_Data=No
THEN    Program_Type_and_Source_Data=New_Development_and_No;
```

```
RULE 390
IF      Program_Type_and_Source_Data=New_Development_and_No AND
        Classified = Yes
THEN    Tmcr_Sections = TMCR2
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 36-01 ...

"
        BCALL tmcr2
        EJECT
BECAUSE
"If the TOs will not have to be classified then the paragraph on
classified TOs can be tailored out of the TMCR 86-01.";

RULE 400
IF      Program_Type_and_Source_Data=New_Development_and_No AND
        Classified = No
THEN    Tmcr_Sections = TMCR4
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 86-01 ...

"
        BCALL tmcr4
        EJECT;

RULE 410
IF      Type_of_Program=Non_Developmental AND
        Source_Data=Yes
THEN    Program_Type_and_Source_Data=Non_Developmental_and_Yes
BECAUSE
"The type of program indicates the probable availability of
commercial manuals and need for Time Compliance Technical Orders.
This information is used in tailoring the TMCR 86-01.";

RULE 420
IF      Program_Type_and_Source_Data=Non_Developmental_and_Yes AND
        Classified=Yes
THEN    Tmcr_Sections = TMCR5
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 86-01 ...

"
        BCALL tmcr5
        EJECT
BECAUSE
"If the TOs will not have to be classified then the paragraph on
classified TOs can be tailored out of the TMCR 86-01.";

RULE 430
```

```
IF      Program_Type_and_Source_Data=Non_Developmental_and_Yes AND
        Classified = No
THEN    Tmcr_Sections = TMCR7
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 86-01 ...

"
        BCALL tmcr7
        EJECT;

RULE 440
IF      Type_of_Program=Non_Developmental AND
        Source_Data=No
THEN    Program_Type_and_Source_Data=Non_Developmental_and_No;

RULE 450
IF      Program_Type_and_Source_Data=Non_Developmental_and_No AND
        Classified = Yes
THEN    Tmcr_Sections = TMCR6
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 86-01 ...

"
        BCALL tmcr6
        EJECT
BECAUSE
"If the TOs will not have to be classified then the paragraph on
classified TOs can be tailored out of the TMCR 86-01.";


RULE 460
IF      Program_Type_and_Source_Data=Non_Developmental_and_No AND
        Classified = No
THEN    Tmcr_Sections = TMCR8
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 86-01 ...

"
        BCALL tmcr8
        EJECT;

RULE 470
IF      Type_of_Program = Class_V_Modification
THEN    Program_Type=Class_V_Mod
BECAUSE
"The type of program indicates the probable availability of
commercial manuals and need for Time Compliance Technical Orders.
This information is used in tailoring the TMCR 86-01.";
```

```
RULE 480

IF      Program_Type=Class_V_Mod AND
        Classified = Yes
THEN    Tmcr_Sections = TMCR9
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 86-01 ...

"
        BCALL tmcr9
        EJECT
BECAUSE
"If the TOs will not have to be classified then the paragraph on
classified TOs can be tailored out of the TMCR 86-01.";

RULE 490
IF      Program_Type = Class_V_Mod AND
        Classified = No
THEN    Tmcr_Sections = TMCR10
        DISPLAY
"
PRINTING THE SUGGESTED TAILORING FOR SECTIONS 2 AND 3 OF THE
TECHNICAL MANUAL CONTRACT REQUIREMENTS 86-01 ...

"
        BCALL tmcr10
        EJECT;

RULE 500
IF       Program_Phase = Concept_Exploration
THEN     First_SOW_Paragraph=No_Inputs

         Rest_of_SOW_and_CDRLs=No_Inputs
         TMCR_Sections=No-Inputs
         DISPLAY
"If your program is in concept exploration then Statement of
Work, Contract Data Requirements List, or Technical Manual
Contract Requirements inputs specifically for TOs are not
needed."
         DISPLAY
" "
         DISPLAY
"In concept exploration, the Logistics Support Analysis program
is the means for gathering logistics data that will be used in
developing the program's TOs in the full scale development and
production phases.  The LSA records that contribute to the
technical order efforts are the C record (operation and
maintenance task summary), the D record (operation and
maintenance task analysis), and the D1 record (personnel and
support requirements)."
         DISPLAY
"
PRESS ANY KEY TO CONTINUE.~";
```

```
RULE 510
IF      Program_Phase = Dem_Val
THEN    First_SOW_Paragraph=No_Inputs
        Rest_of_SOW_and_CDRLs=No_Inputs
        TMCR_Sections=No-Inputs
        DISPLAY
"If your program is in demonstration/validation then Statement of
Work, Contract Data Requirements List, or Technical Manual
Contract Requirements inputs specifically for TOs are not
needed."
        DISPLAY
" "
        DISPLAY
"In demonstration/validation, the Logistics Support Analysis
program is the means for gathering logistics data that will be
used in developing the program's TOs in the full scale
development and production phases.  The LSA records that
contribute to the technical order efforts are the C record
(operation and maintenance task summary), the D record (operation
and maintenance task analysis), and the D1 record (personnel and
support requirements)."
        DISPLAY
"
PRESS ANY KEY TO CONTINUE.~";

RULE 520
IF      Maintenance_Concept = CLS
THEN    First_SOW_Paragraph=No_Inputs
        Rest_of_SOW_and_CDRLs=No_Inputs
        TMCR_Sections=No-Inputs
        DISPLAY
"If your program has contractor logistics support as its
maintenance concept then Statement of Work, Contract Data
Requirements List, or Technical Manual Contract Requirements
inputs specifically for TOs are not needed."
        DISPLAY
" "
        DISPLAY
"A program with contractor logistics support does not require any
TO work since the contractor will be supporting the system/
equipment for the life of the system/equipment.  It is the
contractor's responsibility to see that his people have the
technical manuals needed to operate and maintain the system/
equipment."
        DISPLAY
"
PRESS ANY KEY TO CONTINUE.~";


ASK Program_Name: "What is the name of the program (limited to 40
characters)?";

ASK Type_of_Program: "What type of program is {Program_Name}?";
CHOICES Type_of_Program: New_Development, Class_V_Modification,
Non_Developmental;
```

```
ASK Program_Phase: "What program phase does the contract cover?";
CHOICES Program_Phase: Concept_Exploration, Dem_Val, FSD,
Fsd_and_Prod_Options, Production;

ASK Maintenance_Concept: "What is the maintenance concept for
{Program_Name}?";
CHOICES Maintenance_Concept: Throw_Away, Organic_O_and_I,
Organic_O_and_D, Organic_O_I_and_D, CLS;

ASK Classified: "Is any part of the system classified such that
any TO will have to be classified?";
CHOICES Classified: Yes, No;

ASK Complexity: "What is the best characterization (in terms of
complexity) of the technology used in {Program_Name}?";
CHOICES Complexity: Simple, Moderately_Complex, Highly_Complex;

ASK Source_Data: "Is the system going to be installed on an
aircraft?";

CHOICES Source_Data: Yes, No;

ASK Number_of_TOs: "How many TOs do you anticipate buying?";
CHOICES Number_of_TOs: Less_than_Three,Three_or_More;
```

Appendix B: Sample Output of the Prototype System

The following output was a result of a test case consultation using the prototype for the Mark XV Identification Friend or Foe (IFF) program. The Mark XV IFF contract covers full scale development with production options. The maintenance concept is organic organizational and depot level support. The Mark XV IFF equipment is highly complex and will be installed on many different types of aircraft and ground vehicles. Over twenty TOs will be developed for the Mark XV IFF system and some TOs will be classified. The system is a new development effort. Based on the preceding information about the Mark XV IFF, the prototype system suggested the following Statement of Work inputs, Contract Data Requirements List items, and tailoring of the Technical Manual Contract Requirements 86-01.

STATEMENT OF WORK INPUTS

****

TECHNICAL ORDERS

XX.XX The contractor shall prepare the Technical orders necessary for the organizational level operation and maintenance and the depot level overhaul of the <YOUR PROGRAM NAME> IAW MIL-STD-1790A and Technical Manual Contract Requirement 86-01. The TOs shall require validation, verification, and pre-publication review prior to negatives and reproducible copies being generated. In-process reviews shall be conducted at the 30, 60, and 90 percent completion points to ascertain if the requirements of the military specifications are being met and the technical content is in keeping with the program directions. The TOs shall be prepared to a reading grade level of <SEE MIL-STD-1752 FOR THE READING GRADE LEVEL OF THE AFSC THAT WILL MAINTAIN THE SYSTEM/EQUIPMENT> (AFSC XXXXX) IAW MIL-STD-1752.

TECHNICAL MANUAL PUBLICATION PLANNING

XX.XX The contractor shall prepare and maintain a Technical Manual Publication Plan and submit it for government review and approval before plan acceptance. The plan will be updated as required throughout the life of the contract. (DI-TMSS-80063)

TECHNICAL MANUAL STATUS AND SCHEDULE REPORT

XX.XX The contractor shall prepare a technical manual schedule and status report. (DI-TMSS-80064)

TECHNICAL MANUAL CFAE/CFE NOTICES

XX.XX The contractor shall prepare technical manual CFAE/CFE notices. (DI-TMSS-80067)

REPORT OF TECHNICAL MANUAL COSTS

XX.XX The contractor shall prepare a report of technical manual costs. (DI-TMSS-80068)

TECHNICAL MANUAL VALIDATION

XX.XX The contractor shall validate each manual being developed. The government shall determine the operation site, if required, for the validation effort. The TOMA or his designated representative shall witness the validation. The contractor shall obtain agreement with the government as to the methodologies to be used for each validation effort. The contractor shall prepare a technical manual validation plan and document the methodologies as part of the plan. The technical manual validation plan shall be made a part of the technical manual publication plan. The contractor shall also generate a validation completion report for each technical manual validated. (DI-TMSS-80069, DI-TMSS-80070)

TECHNICAL MANUAL RESEARCH AND ANALYSIS SOURCE DATA

XX.XX The contractor shall prepare source data to the airframe contractor/weapons system manager. (DI-M-6158)

CONTRACT DATA REQUIREMENTS LIST ITEMS

****

TECHNICAL MANUAL PUBLICATION PLAN

DD Form 1423 Block #	Entry
2.	Technical Manual Publication Plan
4.	DI-TMSS-80063
6.	SPO Logistics Office
8.	N
10.	As Required
12.	60 Days After Contract Award
13.	As Required
14.	SPO Logistics Office ALC Using Commands SPO Contracting Office (letter only) SPO Data Management Office (letter only)
16.	Contractor shall provide a technical manual plan IAW the prescribed data item. This manual will include, but not be limited to clear definition of the intended purpose of each manual, delineate the scope of each manual and explain the interface and overlap between or among the manuals. Draft copies of the plan shall be submitted for AF comments prior to submission of the final copy.

TECHNICAL MANUALS STATUS AND SCHEDULE REPORT

DD Form 1423 Block #	Entry
2.	Technical Manuals Status and Schedule Report
4.	DI-TMSS-80064
6.	SPO Logistics Office
7.	LT
8.	N
10.	Monthly
12.	60 Days After Contract Award
14.	SPO Logistics Office ALC Using Commands SPO Contracting Office (letter only) SPO Data Management Office (letter only)
16.	Reference Block 4: Submit complete status and schedule report for all technical orders being delivered under this contract. The specific form for reporting may be of the contractor's choice with approval by the government. Reference Block 13: Date of subsequent submittals is every 30 days until completion of delivery of all technical orders.

TECHNICAL MANUAL CFE/CFAE NOTICES

DD Form 1423 Block #	Entry
2.	Technical Manual CFAE/CFE Notices
4.	DI-TMSS-80067
6.	SPO Logistics Office
7.	LT
8.	AN
10.	As Required
11.	Concurrent w/ AF approval
13.	See Block 16
14.	SPO Logistics Office ALC Using Commands SPO Contracting Office (letter only) SPO Data Management Office (letter only)
16.	CFE notices shall be submitted upon determination that a new piece of equipment requiring a TO is being introduced into the AF inventory or an existing TO requires changes/revisions to be compatible with equipment/modifications being delivered. The notices are to be used primarily for identification of requirements, recommendations, coordination and to assist proper assignment of TO numbers and titles. All TOs addressed on CFAE/CFE notices that are

approved will be procured on a separate contractual instrument. The contractor shall evaluate the commercial manuals by the criteria in MIL-M-7298.

REPORT OF TECHNICAL MANUAL COSTS

DD Form 1423 Block #	Entry
2.	Report of Technical Manual Costs
4.	DI-TMSS-80068
6.	SPO Logistics Office
8.	N
10.	As Required
12.	See Block 16
14.	SPO SPO Logistics Office ALC SPO Contracting Office (letter only) SPO Data Management Office (letter only)
16.	Reference Block 12: Total cost will be established when replying to the RFP. A breakdown of this cost will take place as per schedule established in Block 13. Reference Block 13: Schedule for submissions will be established at the Technical Order Guidance Conference.

TECHNICAL MANUAL VALIDATION PLAN

DD Form 1423 Block #	Entry
2.	Technical Manual Validation Plan
4.	DI-TMSS-80069
6.	SPO Logistics Office
7.	LT
8.	AN
10.	As Required
12.	60 Days After Contract Award
13.	As Required
14.	SPO SPO Logistics Office SPO Contracting Office (letter only) SPO Data Management Office (letter only)
16.	Contractor shall in this plan document the methodologies and procedures for accomplishing the validation effort. These methodologies and procedures will be IAW TO 00-5-1.

TECHNICAL MANUALS VALIDATION COMPLETION REPORT

DD Form 1423 Block #	Entry
2.	Validation Completion Report, Technical Manuals
4.	DI-TMSS-80070
6.	SPO Logistics Office
7.	LT
8.	AN
10.	One Time
12.	See Block 16
14.	SPO SPO Logistics Office SPO Contracting Office (letter only) SPO Data Management Office (letter only)
16.	Contractor shall validate data IAW TO 00-5-1 as applicable. All operating and maintenance procedures will be validated unless waived by the government. The Air Force will witness the contractor's validation. One copy of AFSC Form 11 shall be furnished for each copy of TOs and changes delivered. Verification by the Air Force is required and will be conducted IAW TO 00-5-1, at the contractor's facilities. Validation/Verification will be scheduled on a mutually agreed date, between the Air Force and the contractor, to be determined at the TO Guidance Conference.

TECHNICAL MANUAL RESEARCH AND ANALYSIS SOURCE DATA

DD Form 1423 Block #	Entry
2.	Technical Manual Research and Analysis Source Data
4.	DI-M-6158
6.	SPO Logistics Office
7.	LT
8.	N
10.	As Required
14.	SPO Logistics Office (letter only) SPO Contracting Office (letter only) SPO Data Management Office (letter only) Aircraft ALC or aircraft manufacturer whichever is writing the aircraft TOs
16.	The contractor shall supply the source documentation in accordance with the data item to the aircraft manufacturer for inclusion in the organizational level maintenance tech orders. The contractor shall coordinate with the aircraft manufacturer to determine the source documentation required.

TECHNICAL MANUAL CONTRACT REQUIREMENT 86-01 TAILORING FOR SECTIONS 2 & 3

****

1. Section 2, paragraph 2 - Fill in the TOMA's office symbol.

2. Section 2, paragraph 3 - Change the last sentence to read "The government shall witness the validation." TO 00-5-1 requires the government to witness the contractor's validation.

3. Section 3, paragraphs 1.2.m and 2, - Delete these paragraphs because newly developed equipment will not have commercial manuals available.

4. Section 3, paragraph 17.2 - Fill in the appropriate blank(s) with the word "verification". Delete any statement(s) that is not consistent with the maintenance concept.

Bibliography

1. Berry, Diane C. "The problem of Implicit Knowledge," Expert Systems - The International Journal of Knowledge Engineering, 4(3):144-151 (August 1987).

2. Berry Diane C. and Donald E. Broadbent. "Expert Systems and the Man-Machine Interface," Expert Systems - The International Journal of Knowledge Engineering, 3(4):228-231 (October 1986).

3. Brown, Capt Thomas D. and Capt Dennis R. Lyon. United States Air Force Technical Order Acquisition: What are the Problems and How can They be Corrected. MS thesis, AFIT//GLM/LSM/84S-6. School of Systems and Logistics, Air Force Institue of Technology (AU), Wright-Patterson AFB OH, September 1984 (ADA746931).

4. Buchanan, Bruce G. and Edward H. Shortliffe. Rule-Based Expert Systems. Reading MA: Addison-Wesley, 1984.

5. Citrenbaum, Ronald and others. "Selecting a Shell," AI Expert, 30-39 (September 1987).

6. Defense Systems Management College. The Program Manager's Support System. PMSS Directorate, Department of Research and Information, January 1987.

7. Department of the Air Force. Acquisition Program Management. AFR 800-2. Washington DC: HQ USAF, 9 June 1986.

8. Department of the Air Force. Air Force Technical Order System. AFR 8-2. Washington DC: HQ USAF, 17 April 1987.

9. Department of the Air Force. AF Technical Order System. TO 00-5-1. Washington DC: HQ USAF, 1 January 1986.

10. Department of the Air Force. Integrated Logistics Support Program. AFR 800-8. Washington DC: HQ USAF, 25 June 1986.

11. Freedman, Roy. "Evaluating Shells," AI Expert, 69-74 (September 1987).

12. Frenzel, Louis E, Jr. Understanding Expert Systems. Indianapolis IN: Howard W. Sams & Company, 1987.

13. Goldenthal, Nathan. Expert Systems and Artificial Intelligence. Cleveland OH: Weber Systems, Inc. 1987.

14. Hammell, Robert J. A Prototype Knowledge-based System for Developing Acquisition Strategies (Final Draft). MS Thesis, AFIT/GCS/ENG/88M. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH, November 1987.

15. Harmon, Paul and David King. Expert Systems: Artificial Intelligence in Business. New York NY: John Wiley & Sons, 1985.

16. Hayes-Roth, Frederick and others. Building Expert Systems. Reading MA: Addison-Wesley, 1983.

17. Hoffman, Robert R. "The problem of Extracting the Knowledge of Experts from the Perspective of Experimental Psychology," AI Magazine, 53-67 (Summer 1987).

18. Mittal, Sanjay and Clive L. Dym. "Knowledge Acquisition from Multiple Experts,"AI Magazine, 32-36 (Summer 1985).

19. Manguia, Arthur A. "Technical Order Acquisition Cycle," Air Force Journal of Logistics, 24-26 (Summer 1985).

20. Munguia Arthur M. and William Epperson. "Taking Aim - Reducing the Cost of Technical Manuals and Technical Orders," Logistics Spectrum 21:25-27 (Winter 1987).

21. Nguyen, Tin A. and others. "Knowledge Base Verification," AI Magazine, 69-75 (Summer 1987).

22. Olson, Judith R. and Henry H. Reuter. "Extracting Expertise from Experts: Methods for Knowledge Acquisition," Expert Systems - The International Journal of Knowledge Engineering, 4(3):152-168 (August 1987).

23. Parnell, Lt Col Gregory S. and Capt Thomas Triscari. "Intelligent Decision Systems for Systems Acquisition Management at Aeronautical Systems Division," (Draft Report). Report to Aeronautical Systems Division Commander. AFIT, Wright-Patterson AFB OH, 5 October 1987.

24. Prerau, David S. "Knowledge Acquisition in the Development of a Large Expert System," AI Magazine, 43-51 (Summer 1987).

25. Prerau, David S. "Selection of an Appropriate Domain for an Expert System," AI Magazine, 26-30 (Summer 1985).

26. President's Blue Ribbon Commission on Defense Management. "A Quest for Excellence," Report to the President. Washington DC, June 1986.

27. Richer, Mark H. "An Evaluation of Expert System Development Tools," Expert Systems - The International Journal of Knowledge Engineering, 3(3):166-183 (July 1986).

28. Swartout, William R. and Stephen W. Smoliar. "On Making Expert Systems More Like Experts," Expert Systems - The International Journal of Knowledge Engineering, 4(3):196-207 (August 1987).

29. Thurman, Lt Gen William. "Artificial Intelligence," Letter to ASD Deputate Commanders. HQ ASD, Wright-Patterson AFB OH, 23 July 1987.

30. Waterman, Donald A. A Guide to Expert Systems. Reading MA: Addison-Wesley, 1986.

VITA

Captain James F. Harvell [redacted] Following graduation, Captain Harvell attended Troy State University in Alabama. For his undergraduate work, he received a Bachelor of Science degree in Computer Science and Mathematics in 1984. Immediately following graduation, Captain Harvell was commissioned into the USAF through the ROTC program at Troy State. He entered active duty in June 1984. His first assignment was as an Integrated Logistics Support Manager for the Precision Location Strike System and Advanced Tactical Air Reconnaissance System at Aeronautical Systems Division, Wright-Patterson AFB, Ohio. In June 1987, Captain Harvell entered the School of Systems and Logistics, Air Force Institute of Technology, as a graduate student.

[redacted]

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ADA201490

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GSM/LSY/88S-11	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Systems and Logistics 6b. OFFICE SYMBOL (If applicable) AFIT/LSY	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB OH 45433-6583	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

A PROTOTYPE EXPERT SYSTEM FOR TECHNICAL ORDER ACQUISITION (UNCLASSIFIED)

12. PERSONAL AUTHOR(S)

James F. Harvell, B.S., Capt, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
MS Thesis	FROM ______ TO ______	1988 September	122

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Artificial Intelligence, Acquisition, Technical Writing, Logistics Support
12	09		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Chairman: Thomas Triscari, Jr., Major, USAF
Associate Professor of Systems Management

This research resulted in a prototype expert system for technical order (TO) acquisitio The prototype was built using VP-EXPERT and runs on the Zenith Z-248 microcomputer. The output of the prototype system consists of the contractual documentation for a TO program: verbiage for the Statement of Work, Contract Data Requirements List items, and tailoring of sections two and three of the Techncial Manual Contract Requirements 86-01.

The research focused on the following six areas: the applicability of expert system technology to TO acquisition tasks; The required resources, participants, goals, and problem characteristics for the prototype; the key concepts and relations of the selected domain of TO acquisition; the appropriate knowledge representation scheme and development tool; the required data structures and control strategies; and the competency and utility of the syste

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Thomas Triscari, Jr., Major, USAF	22b. TELEPHONE (Include Area Code) (513) 255-4845 / 22c. OFFICE SYMBOL AFIT/LSY

Approved for public release IAW AFR 190-1.

WILLIAM A. MAUER 17 Oct 88

Associate Dean

School of Systems and Logistics

Air Force Institute of Technology (AU)

Wright-Patterson AFB OH 45433